

SUPER

Anno 2
n. 6 - Giugno 1985

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

6

sinc

SUPERSINC
la rivista per gli utenti
dei prodotti
SINCLAIR

Un pubblicazione della

J.soft EDITRICE

In collaborazione con

GRUPPO
EDITORIALE
JACKSON



**TUTTI ASSEMBLER
APPASSIONATAMENTE:
GUIDA AI MIGLIORI
PROGRAMMI
SUL MERCATO**

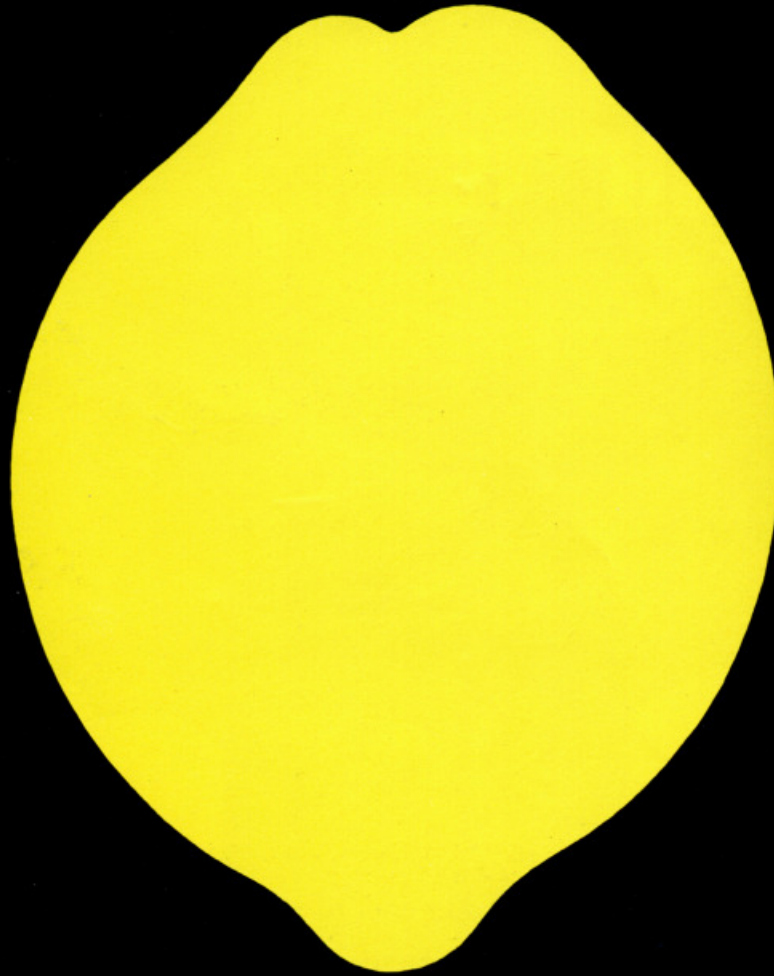
**SALTAMASSI
E SLALOM
PER ZX81 1K**

**SPECPAINT
UN GRANDE GIOCO
PER TUTTI!**

**PROSEGUE
IL CORSO DI L/M**

**IN OMAGGIO
IL COMPATTATORE
DI SCREEN \$**

Per la sete di soft



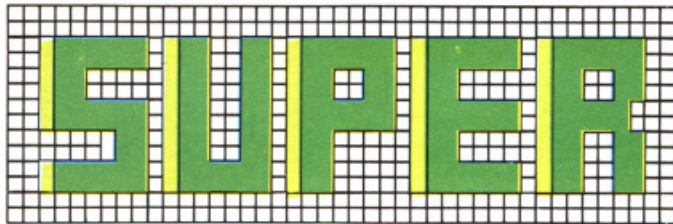
Zeta 2

Il nuovo Lemon II è ancora più potente: nuove soluzioni tecniche e il drive da 160K incorporato consentono a questa macchina prestazioni eccezionali. È possibile infatti utilizzare tutti i migliori programmi professionali: package di gestione, data base, foglio elettronico, word processing, grafica, archiviazione... Lemon II è compatibile, ha un prezzo accessibile, garanzia e rete di assistenza tecnica. È l'occasione per avere subito un vero computer professionale.



Organizzazione di vendita:
Torino 011.337744
Milano 02.4232437
Bologna 051.223714
Firenze 055.372281
Roma 06.5420305/5423716
Caserta 0823.460469
Catania 095.416560

LEMON II
il grande compatibile



5 EDITORIALE
POINT

L'IMPORTANTE È NON
ESAGERARE
di **Pietro Dell'Orco**

7 READ
& WRITE
LA POSTA DEI LETTORI

9 OPEN

TUTTI ASSEMBLER
APPASSIONATAMENTE
di **Marcello Spero**

13 LA CULTURA ^{DS} INFORMATICA
...then

AVVENTURA
NELL'ADVENTURE
di **Brian Robb**
trad. e adatt.
di **Marcello Spero**
Prima parte

17 REM:HW

IL TERMOSTATO BEN
TEMPERATO
di **Marcello Spero**
Seconda parte

23 RANDOMIZE

L'ASSEMBLY
ASSIEME
di **Marcello Spero**
Terza parte

29 LOAD

29 RADAR
di **Bill Lythgoe**
trad. e adatt.
di **Angela Spinazzola**

33 ZANZARE
di **Sam French**
trad. e adatt.
di **Angela Spinazzola**

37 RACCOLTA NEL
FRUTTETO
di **Angus Lavery**
trad. e adatt.
di **Angela Spinazzola**

41 SINCRONIA
di **D.O. Wright**
trad. e adatt.
di **Angela Spinazzola**

47 IL CASTELLO
STREGATO
di **Babar Abidi**
trad. e adatt.
di **Paolo Maffei**

51 SPECPOINT
di **Carlo Panzalis**

61 MOTORE A DUE
TEMPI
di **Tim Rose**
trad. e adatt.
di **Angela Spinazzola**

63 SLALOM
di **Andrew Norton**
trad. e adatt.
di **Guido Bertolotti**

64 SALTAMASSI
di **Luuk Hilhorst**
trad. e adatt.
di **Guido Bertolotti**

65 TAVOLOZZA
di **Dino Citterio**

74 PRINT

76 RECENSIONI LIBRI
ZX CLUB

81 I CLUB SINCLAIR
input-output

PICCOLI ANNUNCI

J. soft s.r.l.
**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228-68.37.97
68.80.841/2/3

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Lucio Bragagnolo
Marcello Spero

**HANNO COLLABORATO A
QUESTO NUMERO**
Guido Bertolotti
Dino Citterio
Paolo Maffei
Carlo Panzalis
Angela Spinazzola

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonatti

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE:**
Tribunale di Milano n° 199
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e
l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 Milano
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta) L. 38.500
(11 numeri senza cassetta) per l'estero:
L. 110.000 (11 numeri con cassetta)
L. 50.000 (11 numeri senza cassetta)
I versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5
20124 MILANO

mediante emissione di assegno
bancario, vaglia o
utilizzando il c/c postale
n. 19445204.
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli
© TUTTI I DIRITTI DI
RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI
SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA
ZX COMPUTING - ARGUS
SPECIALIST
PUBLICATION LTD
1 GOLDEN SQUARE LONDON
W 1R3AB



Rivista associata
all'Unione
Stampa
Periodica
Italiana

Piccola guida per l'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati: in questo modo speriamo di avere ridotto al minimo la possibilità di errori. Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da approntare in uno dei numeri immediatamente successivi della rivista.

Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.

Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.

Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).

In entrambi i casi per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante), premendo contemporaneamente i tasti CAPS-SHIFT e 9.

Nei nostri listati i caratteri grafici predefiniti sono indicati da una Q seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.

Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.

Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico, premendo contemporaneamente il tasto CAPS-SHIFT.

Ad esempio quando, si trova la codifica {G2} occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.

Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.

I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.

Quando in un listato viene presentata, ad esempio, una A sottolineata occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.

Quando leggete	Premete	Vedrete
{G1}		
{G2}		
{G3}		
{G4}		
{G5}		
{G6}		
{G7}		
{G8}		
{SG1}		
{SG2}		
{SG3}		
{SG4}		
{SG5}		
{SG6}		
{SG7}		
{SG8}		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

Quando leggete	Premete	Vedrete
<u>A</u>		Simbolo grafico definito nel programma in uso.
<u>B</u>		
<u>C</u>		
<u>D</u>		
<u>E</u>		
<u>F</u>		
<u>G</u>		
<u>H</u>		
<u>I</u>		
<u>J</u>		
<u>K</u>		
<u>L</u>		
<u>M</u>		
<u>N</u>		
<u>O</u>		
<u>P</u>		
<u>Q</u>		
<u>R</u>		
<u>S</u>		
<u>T</u>		
<u>U</u>		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

POINT

**L'importante è
non esagerare**

D'accordo, c'è la passione, c'è la voglia di fare e imparare sempre cose nuove, ma se la cosa diventa un fatto "patologico", questo non va bene. Di giorno ci sono le attività a cui ciascuno di noi deve necessariamente sottendere, la notte, salvo eccezioni (leggi guardiani notturni), è necessario dormire.

Dunque, il tempo da dedicare al computer dev'essere recuperato dal cosiddetto "tempo libero" che, per quanto sopra, subentra la costrizione di dover operare sul calcolatore solo per brevi periodi di tempo, che a volte condizionano la riuscita di quanto si desidera fare.

Quando si pone questa condizione gli utenti solitamente si trasformano in una sorta di "zombies", che passano ogni ora disponibile (comprese quelle notturne) appiccicati alla tastiera, cercando di far digerire alla macchina quanti più byte possono.

Se questo modo di agire diviene prassi ricorrente, le implicazioni sono quasi sempre poco piacevoli. Bocciature, liti furibonde con il coniuge, insonnia cronica e via dicendo.

La sostanza di tutto il discorso si riassume in un sol motto: non esagerare! Senza contare poi che, di solito, quando ci si accanisce con un programma che non vuole girare a dovere, è molto difficile porvi rimedio se la mente non è riposata.

L'importante è non lasciarsi coinvolgere al punto da trasformare il computer nell'unico interesse della propria vita, dedicandogli pertanto tutte le risorse di tempo e di energia di cui l'individuo dispone.

Il consiglio dunque è uno solo: misura e moderazione che, oltre ad evitare spiacevoli inconvenienti, faranno maggiormente apprezzare il tempo, anche se poco, speso sulla tastiera del computer a dare legittimo sfogo alla propria passione.

Pietro Dell'Orco

Per collaborare a SUPERSINC

La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Horance Goes.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e sopra-

tutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare, per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE, piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24x36, o 6x6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

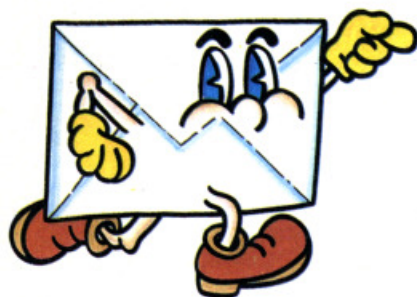
15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviare idee e programmi a:
SUPERSINC
Viale Restelli, 5
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

READ & WRITE



La posta
dei lettori

Bug concettuali

Spett. redazione,

sono utente di uno Spectrum 16K e mi sto cimentando nella creazione di un mio programma. Proprio mentre pensavo di avere risolto tutti i guai e di avere steso la versione definitiva, è sorto un problema... allego i francobolli per la vostra risposta.

Alessio Battisti - Roma

Di regola non rispondiamo direttamente a casa dei lettori, anche se possono esserci delle eccezioni, che comunque non contemplano il caso in questione. Riguardo al problema specifico ipotizziamo che le istruzioni di INKEY\$ siano esterne al loop di movimento dell'aereo; ma le spiegazioni contenute nella sua lettera sono largamente insufficienti. Ci riscriva inviandoci il listato del programma, se non addirittura una cassetta, unitamente alle spiegazioni del caso.

Ispezionando il LISP

Dispongo di un ottimo interprete LISP per Spectrum. Ho avuto la cassetta da amici romani, che non disponevano del manuale, e quest'ultimo (fotocopie incomplete) da amici torinesi. L'interprete, una volta caricato, inizia dichiarando

SpecLisp 1.2
(c)1983 Serious Software
Made in England

Il manuale invece annuncia che sullo schermo apparirà

specLISP 1.2
(c) 1982 K. P. Stone 3' c
Made in England

Nonostante queste differenze, si tratta dello stesso interprete SpecLISP 1.2, come ho potuto verificare applicando mi-

nuziosamente il manuale all'interprete. Entrambi i nominativi delle case di software risultano sconosciuti presso i venditori. né compaiono su riviste italiane o straniere.

Vi ho scritto nella speranza che possiate fornirmi qualche indicazione: sarebbe mio desiderio poter acquistare l'interprete e la documentazione relativa, al di là del fatto di disporre già tramite copia di copia di copia di...

Cordiali saluti e grazie.

Pietro Terna - Torino

Egregio signor Terna, confermiamo il suo giudizio sullo SpecLISP 1.2, programma veramente notevole. Purtroppo temiamo che riuscire a trovare in Italia una confezione originale del programma sia cosa difficile, vista la scarsa propensione dei rivenditori a commerciare software di questo tipo. Approfittando quindi dell'occasione per sollecitare i lettori in possesso di informazioni adeguate a segnalare, le forniamo per ora l'indirizzo della Serious Software, che dovrebbe essere in grado di soddisfare la sua richiesta:

Serious Software
7 Woodside Road
Bickley, Kent BR1 2ES

In più, anticipiamo che in uno dei prossimi numeri, nell'ambito della serie sui linguaggi alternativi già inaugurata dal Pascal, IF...THEN ospiterà un articolo di carattere introduttivo sul LISP.

Stranezze del jet set

Se si scrive questa riga di programma:

```
25 POKE 23672,0: POKE 23673,0:  
POKE 23674,0
```

e si esegue un MERGE con il programma caricatore di "Jet Set Willy", dando RUN

e caricando il resto del gioco il programma andrà in autorun, senza richiedere il codice di entrata.

Elio Solino - Brugherio (MI)

Ma ne sei proprio sicuro? Noi abbiamo provato per più volte su tutti i nostri Spectrum (compreso quello di casa), senza risultato. Provvederemo a documentarci... intanto, ecco un trucco - sempre relativo a "Jet Set Willy" - di altro genere. Esso consiste nell'aggiungere al solito programmino caricatore una riga

```
25 POKE 35899,0
```

Questo accorgimento permette di disporre di un numero infinito di vite.

Programmi per il futuro

I programmi contenuti nella rivista sono decisamente banali e privi di interesse. Molto interessanti, invece, sono le rubriche IF...THEN, REM:HW, RANDOMIZE (eccezionale l'articolo su "Atic Atac")...

Beppe Fasolis - Asti

Non è vero che i nostri programmi siano del tutto "banali e privi di interesse": anche trascurandone le tematiche, essi possono rivelarsi fonte inesauribile di routine e trucchi utili a tutti i programmatori. In più, alcuni dei listati pubblicati finora sono di livello paragonabile a quello di qualsiasi medio programma commerciale (vedi "Gizzy" per ZX81, o "Colliders", o "Galattoidi", tutti facenti largo uso del linguaggio macchina). È chiaro che, parlando di giochi scritti in BASIC, un raffronto con "Sabre Wulf" o "Atic Atac" non sarebbe neanche serio; ma è anche vero che alcuni programmi, come ad esempio "Risico", "Micropoli", o "Spectrum dietologo", sono praticamente inediti e reggono il

confronto col software commerciale in maniera più che dignitosa. Che dire poi di utility come "Renumber" o "42 colonne"? Quanto a RANDOMIZE, forse proprio in questo numero (o nel prossimo) potete vedere un altro articolo in stile "Atic Atac", dedicato ad "Alchemist", sempre ad opera del bravissimo Carlo Panzalis. E altri seguiranno!

Quesiti squisiti

Caro SUPERSINC,

mi chiamo Michele e vi scrivo per la prima volta. Vorrei chiedervi due cose:

1) il gioco "Formula 1" scritto per lo ZX81 si può riscrivere e modificare per lo Spectrum?

2) Perché non pubblicate classifiche di giochi o utility?

Sperando di avere una risposta vi saluto.

**Michele Bertagnolo
Besnate (VA)**

Caro Michele,

1) finora nessuno dei nostri collaboratori ci ha pensato: tuttavia l'idea è certamente realizzabile;

2) perché la nostra classifica ideale è quella fatta dai lettori: non vogliamo imporre i nostri pareri, ma pubblicare i loro, anzi, i vostri. Perché non ci mandi la tua classifica personale?

Elle emme

La necessità principale di noi utenti Spectrum è quella di conoscere il linguaggio macchina. Impegnatevi su questa strada: non chiedo altro...

**Giovanni Florio
Palo del Colle (BA)**

Sulla rivista appare già da tempo un corso di linguaggio macchina e cominciano ad aumentare i programmi contenenti routine scritte in l/m. Naturalmente, rimarrà sempre un ampio spazio per i puristi amanti del solo BASIC, ma non è assolutamente nostra intenzione trascurare questo importante aspetto della programmazione.

Publicatemi!

Desidero proporvi alcune domande che penso possano interessare molti spectrumisti (preferiamo "sinclairisti": suona meglio! N.d.R.). Spero in una vostra risposta. Mi raccomando: se le domande sono troppe, scartatene pure qualcuna, ma pubblicatemi! Anche in ritardo, ma pubblicatemi!

1) Con che ordine e con quale criterio rispondete alle lettere dei lettori?

2) Come si fa a proteggere i programmi dal BREAK?

3) Giochi come "Atic Atac" o "Manic Miner", dato che cambiano continuamente di ambientazione, possono essere considerati avventure?

4) Consiglierei di istituire una pagina de-

dicata alla pubblicazione dei bug, come quello che in "Manic Miner" permette di cambiare schermo.

5) Cosa pensate delle riviste "su cassetta"?

6) Cosa ne pensate dei programmi per Spectrum trasmessi da RAI 3?

**Carlo Morandini
Piamborno (BG)**

Di fronte a tanta insistenza non si può non rispondere! Dunque:

1) l'ordine è strettamente cronologico, mentre il criterio è praticamente quello del grado di interesse generale della lettera, anche se facciamo diverse eccezioni. Bisogna anche che i temi affrontati siano nuovi per la rivista: a tutti coloro che hanno scritto in cerca di consigli per finire "Atic Atac" sicuramente non verrà data risposta, visto che l'argomento è stato sviscerato su SUPERSINC numero 5 in tutti i suoi aspetti.

2) L'unico modo per proteggere perfettamente un programma dal BREAK è quello di scrivere una routine appropriata in linguaggio macchina, attaccandola al programma stesso. In parole povere, si tratta di questo: nella ROM dello Spectrum esiste una routine che, in caso di errore (il BREAK viene appunto trattato come tale), interrompe l'esecuzione del programma, visualizzando un messaggio appropriato, da OK a "Nonsense in BASIC" a "Variable not found", per citarne alcuni dei più comuni. La nostra routine ha quindi l'incarico di far sì che in una situazione simile l'interprete BASIC non salti a routine contenute in ROM, bensì, per esempio, a una linea del programma, o a un nostro messaggio di errore in esso contenuto, o addirittura provochi il reset dello Spectrum.

È chiaro che vanno considerate tutte le possibilità e che è opportuno prevedere comunque un errore "normale", per non essere costretti a spegnere il computer per liberarsi del programma (ad esempio uno "STOP in INPUT"). Questo per ciò che riguarda la teoria, sia pure in termini molto grossolani. Della pratica riparleremo...

3) Quella della classificazione dei giochi è questione controversa e delicata, nonché un tantino accademica. A nostro parere "Atic Atac" è effettivamente un'avventura, mentre "Manic Miner" è solamente arcade game. È vero che anche nel secondo esistono numerosi ambienti in cui operare, ma questi sono "sequenziali" (ossia per raggiungere lo schermo 12 bisogna avere superato gli altri 11) e non richiedono altro che abilità manuale unita a un po' di riflessione, consistente nel trovare il pattern più conveniente per risolvere certe difficoltà. La successione degli schermi di "Manic Miner" non è molto diversa da quella di "Chuckie Egg" o di un qualsiasi Pacman, mentre in "Atic Atac" ci sono elementi che costringono all'adozione di una strategia decisamente più comples-

sa - per esempio il bisogno di crearsi una mappa -, uniti ad altri che all'inizio del gioco sono completamente sconosciuti, come i passaggi segreti o la posizione di premi essenziali alla soluzione dell'avventura.

4) Al momento questa funzione è svolta da READ & WRITE, pensiamo in maniera soddisfacente.

5) Iniziative buone da un punto di vista divulgativo ed editoriale; diventano però estremamente dannose per tutto il mercato del software nel momento in cui vivono pirateggiando giochi originali inglesi, di cui vengono semplicemente modificati lo schermo di presentazione originale e qualche messaggio all'interno del programma. I prezzi del software non sarebbero così alti, se le software house non venissero danneggiate così disonestamente e in questa misura. In più, questo fenomeno scoraggia i programmatori dalla realizzazione di nuovi prodotti (quanti programmatori, in Italia, sono in grado di vivere scrivendo software per lo Spectrum? Nessuno. In Inghilterra se ne contano a decine). Infine, se la necessità di proteggere il proprio lavoro dai ladri non fosse così pressante, non esisterebbero i meccanismi di protezione che tutti conosciamo e che rendono di fatto impossibile a un principiante la realizzazione di una copia di riserva per uso personale. E se questo non bastasse, cosa penseremmo di una persona che, oltre a far guadagnare meno noi, si arricchisce rubando il nostro lavoro e firmandolo come se lo avesse fatto lui?

6) È un'ottima iniziativa, purtroppo limitata dalle difficoltà di registrazione dei programmi, dovute alla non sempre sufficiente qualità del segnale. Ad ogni modo riteniamo che il prossimo futuro informatico sarà dominato, più che da queste iniziative, dalle banche dati a cui è possibile collegarsi via telefono (come già accade in USA e Gran Bretagna).

Spectrum e spettri

Caro SUPERSINC,

sono un appassionato di b.f. e di audio in generale, e vorrei sapere se è possibile adoperare lo Spectrum 48K come analizzatore di spettro audio, come fare per realizzare l'interfaccia e come realizzare il relativo programma. Vi prego di aiutarmi! Nell'attesa, porgo distinti saluti.

Daniele Meini - Legnano (MI)

Egregio signor Meini, in uno dei prossimi numeri troverà, come omaggio incluso nella cassetta, un programma realizzato dal Gruppo Utilizzatori Computer Sinclair Napoli, dalle prestazioni discrete e che non necessita di interfacciamenti particolari. Le consigliamo, per avere maggiori informazioni, di rivolgersi direttamente al Gruppo stesso, del quale abbiamo già pubblicato (e pubblicheremo) numerose volte tutti i dati necessari.



OPEN

Tutti Assembler appassionatamente

di **Marcello Spero**

Un'esplorazione tra i più interessanti programmi pensati per oltrepassare la frontiera del BASIC.

Assembler, editor, disassembler, monitor. Dietro questi vocaboli "inquietanti" si nascondono alcuni degli strumenti più importanti per chiunque, esperto o meno esperto, si cimenti nella programmazione in linguaggio macchina, o più in generale cerchi di sfruttare a fondo le possibilità del computer di cui dispone.

L'idea di prendere in esame questo tipo di software si è affermata, visto l'interesse suscitato dal nostro corso di linguaggio macchina. Un tale interesse testimonia la crescita di una "nuova generazione" di utenti, che non si accontentano del ruolo puramente passivo di utilizzatori, ma vogliono creare qualcosa di personale, piegando la macchina alle loro esigenze. Che questo qualcosa, poi, sia un gioco o uno strumento di lavoro, poco importa.

Come abbiamo detto, non si tratta di qualcosa che possa interessare quanti concepiscono (e pretendono) il computer unicamente come console per videogame, rifiutando qualsiasi sforzo volto ad imparare e progredire. Ma d'altronde nella società di oggi (e di domani) di posto per gente così ce ne sarà sempre meno, e quindi...

Prima di tutto cerchiamo di capire di cosa si tratta.

Tutti i microcomputer, e quindi anche lo Spectrum, hanno "a bordo" un unico linguaggio, e cioè il BASIC. Essendo questo rivolto, come tutti i linguaggi ad alto livello, all'uomo e non alla macchina, le sue possibilità di agire su strutture come byte, aree di memoria, puntatori, registri sono del tutto marginali.

Una situazione di questo tipo va benissimo quando si fa un uso "proprio" del BASIC. Se invece per le ragioni più va-

rie (ce ne siamo occupati, e ce ne occuperemo ancora, nel corso di linguaggio macchina) si sconfinano in un uso "improprio" di questo linguaggio, muoversi a furia di PEEK e POKE può diventare insostenibile.

Con questo si vuol dire che il BASIC è uno strumento ottimo per scrivere programmi chiari e leggibili che facciano riferimento alle possibilità "normali" della macchina, ma diventa pesantissimo se usato per agire direttamente sulle strutture fisiche del computer.

Quando si arriva a questo stadio, spinti dall'esigenza di far fare al computer quello che normalmente non potrebbe fare (scrolling laterale, animazione ad alta risoluzione, calcoli veloci, caricamento rapido da cassetta,...) il linguag-

gio macchina è la meta obbligata.

Gli inizi di tutti gli aspiranti programmatori in l/m sono molto simili: traduzione "manuale" del linguaggio Assembly nei codici macchina (con tanti errori...), loro caricamento a suon di POKE, verifica di lunghe liste di numeri ottenute con delle PEEK.

Il problema maggiore, comunque, non è il caricamento, specialmente finché si tratta di qualche decina di codici. Il problema vero, dicevamo, sorge nel momento in cui ci si accorge (e purtroppo capita nove volte su dieci...) di aver commesso qualche errore. La correzione (il famoso "debugging") di programmi in linguaggio macchina, avendo a disposizione le sole PEEK e POKE più carta e matita è un'esperienza non pia-



cevole, come può confermare chi ci ha provato.

Un altro aspetto poco praticabile senza strumenti adatti è la modifica, la fusione, i vari "taglia e incolla" che possono essere utili per impiegare vecchie routine per nuovi scopi. Persino il semplice spostamento da una zona di memoria all'altra del medesimo blocco in I/m è un'impresa densa di problemi: basti pensare al reperimento di tutti i salti assoluti ed alla loro correzione...

Per risolvere tutti questi problemi, e rendere la programmazione in linguaggio macchina (o meglio in Assembly, visto che è questo il nome che prende il I/m quando è espresso non in forma di numeri, ma di sigle letterali) piacevole e facile per lo meno quanto lo è quella in BASIC, sono stati creati gli strumenti che abbiamo elencato in apertura.

Possiamo dividerli in due categorie, di cui parleremo separatamente:

- Assembler, con o senza editor proprio
- Disassembler, normalmente abbinati ad un monitor

Assembler

Traducono il linguaggio Assembly nei codici macchina comprensibili al computer.

Si tratta in sostanza dello stesso lavoro che possiamo fare noi con carta e matita, ma con alcune sostanziali differenze.

La prima riguarda la velocità. Un assembler mediamente è in grado di tradurre (assemblare) decine e decine di mnemoniche (così si chiamano le "istruzioni" Assembly) al secondo.

La seconda riguarda, ovviamente, gli errori. Questi sono, come tutti sanno, una prerogativa squisitamente umana; un Assembler non può commettere errori di traduzione.

C'è da dire infine che tutti gli Assembler consentono l'uso di etichette, per indicare il punto di arrivo dei vari salti. In questo modo al programmatore viene evitato qualsiasi tipo di calcolo.

Ogni ditta offre poi la sua gamma personale di possibilità extra, di cui parleremo più avanti. Per il momento è sufficiente dire che alcune di queste opzioni possono essere così interessanti da giustificare da sole l'acquisto di un Assembler invece di un altro.

Due parole vanno spese per gli editor. Un editor è l'insieme di comandi, opzioni e possibilità varie che rendono più o meno facile la stesura di un blocco di testo. Per testo si intende non solo un brano letterario, ma anche un programma.

Un esempio di editor, piuttosto buono, che tutti voi avete davanti continuamente è quello incorporato nello Spectrum, che vi consente di scrivere, cancellare, spostarvi all'interno di una linea, richiamare una linea qualsiasi, listare a piacere tutto il programma o parte di esso.



Per l'esattezza, un editor del tipo di quello dello Spectrum si dice "screen editor", poiché l'esecuzione dei suoi comandi avviene direttamente sullo schermo per mezzo di un cursore. Dal lato opposto ci sono i "line editor", in cui i comandi sono espressi con sigle e numeri, e l'utilizzatore non ha quella sensazione di chiarezza ed immediatezza propria degli "screen".

Molti Assembler dispongono di un editor proprio, diverso da quello cui siamo abituati. Se da un lato questo è un vantaggio (normalmente, infatti, le possibilità sono maggiori), dall'altro è richiesto un certo tempo per "prendere la mano".

Altri assembler, proprio per questo motivo, fanno uso dell'editor dello Spectrum. Normalmente, in questi casi il testo del programma Assembly viene scritto in linee REM. Ci si trova insomma di fronte ad un programma BASIC composto da sole linee REM, e quindi tutti i normali comandi (cursors, delete, list) funzionano nel modo solito.

In ambedue i casi le opzioni supplementari si sprecano, alcune utili, altre meno. Sempre presenti, e molto utili (chissà perché non sono "di serie" sullo Spectrum), AUTO e FIND. Si tratta della numerazione automatica di linea e della ricerca automatica di una particolare parola o sequenza di caratteri, e loro eventuale sostituzione con altri.

Monitor/Disassembler

Il loro compito è più o meno opposto a quello degli Assembler. Si tratta infatti di strumenti software per operare sul linguaggio macchina già tradotto ed inserito in memoria.

Esistono sul mercato microcomputer che dispongono fin dalla nascita di strumenti di questo tipo. Non è comunque il caso dello Spectrum.

Le funzioni più tipiche di un monitor sono:

- Visualizzazione e possibilità di modifica del contenuto di una zona qualsiasi della RAM
- Spostamento di un blocco di dati da una zona all'altra della memoria
- Riempimento di un dato numero di byte con un carattere specificato
- Aggiunta o cancellazione di byte in un blocco di I/m
- Inserimento di "breakpoint" all'interno di una routine in I/m, in modo da poterne provare solo una parte
- Visualizzazione del contenuto dei vari registri della CPU
- Inserimento in memoria di blocchi di testo, i cui caratteri vengono automaticamente convertiti nei codici
- Conversione decimale/esadecimale, e viceversa, di numeri
- Disassemblaggio di blocchi di I/m.

L'unico punto che necessita di una spiegazione è l'ultimo, e cioè il disassemblaggio.

Si tratta della caratteristica più importante. Consiste nella possibilità di ritradurre in mnemoniche i codici numerici. Questo vale sia per quelli tradotti da noi, manualmente o con un Assembler, che per quelli di altri (giochi in I/m), e persino per quelli in ROM.

L'unica cosa da tener presente quando si usa un qualsiasi disassemblatore è l'indirizzo di inizio. Basta infatti partire spostati di un indirizzo per ottenere una traduzione completamente sballata. Un esempio chiarirà questo concetto.

Supponiamo di avere in memoria la sequenza

33, 32, 00, 35, 117

Se la disassembliamo a partire dalla prima cifra, otteniamo:

ld hl,32	33	32	00
inc hl	35		
ld(hl),l	117		

Se invece la disassembliamo a partire dalla seconda cifra, avremo:

jr nz,00	32	00
inc hl	35	
ld(hl),l	117	

che, come vedete, è molto diverso. Un'altra cosa da tener presente è che il Disassembler non è in grado di distinguere i blocchi di dati dalle sequenze di programma. È facile quindi, se non si sta attenti, interpretare come mnemoniche dei codici che in realtà sono dati inseriti nel programma.

Ora che ci siamo fatti un'idea di Assembler, Disassembler, Monitor, Editor possiamo dare uno sguardo a quello che offre il mercato.

La varietà di prodotti è, specialmente per lo Spectrum, davvero enorme, e sarebbe impossibile passare in rassegna ogni singolo package. Si è pensato perciò di suddividere questi prodotti in categorie diverse, ed esaminare di ciascuna categoria il prodotto più rappresentativo. In questo modo verrete a disporre di un metro di paragone per qualsiasi prodotto del genere abbiate l'opportunità di vedere.

Nella prima categoria poniamo i package di minori pretese, ma d'altra parte più semplici da usare. Gli Assembler di solito usano l'editor dello Spectrum. Un buon esempio di questo tipo è costituito dallo ZX SPECTRUM MACHINE CODE ASSEMBLER, prodotto dalla Mc Graw-Hill e distribuito dalla ACS software. Non sono presenti opzioni extra per l'editor, mentre l'Assembler dispone del numero minimo di pseudo-istruzioni necessarie al suo funzionamento. Si tratta di

go e finish

per l'inizio e la fine del listato Assembly,

org

per definire l'indirizzo in cui si troverà il l/m pronto a funzionare,

defb
defw
defs

per l'inserimento nel listato di singoli byte, coppie di byte (word) e stringhe,

equ

per attribuire un valore ad un'etichetta.

Si tratta, come abbiamo detto, del mini-



mo necessario al buon funzionamento dell'Assembler, ma per chi non ha eccessiva dimestichezza con questo genere di cose la semplicità "francescana" del Mc Graw-Hill può essere un grosso vantaggio.

Appena un gradino più su troviamo Assembler dello stesso tipo (editor dello Spectrum, numero minimo di pseudo-istruzioni), che dispongono però di un "tool kit", cioè una serie di strumenti atti a rendere più rapida e agevole la programmazione.

Per tutti citiamo l'ASTRON, della DK'Tronics, che ci sembra uno dei migliori. Il suo tool kit dispone di parecchie cosette interessanti, come

alter

per sostituire determinati gruppi di caratteri con altri,

bytes

per conoscere quanto spazio è rimasto in memoria

copy

per duplicare un numero qualsiasi di linee di programma

delete

per cancellare un blocco di linee

find

per trovare una determinata sequenza di caratteri

move

per spostare un blocco di linee da un punto all'altro del programma

renumber

per rinumerare tutto o parte del programma, definendo il nuovo numero iniziale e la spaziatura

sequence

per la numerazione automatica delle linee, che appaiono già complete di REM

verify

quando è inserito lista tutte le modifiche effettuate da alter.

Questo tool kit è disponibile sia nella versione 16K che nella 48K, ma in quest'ultima può risiedere in memoria contemporaneamente all'Assembler, mentre nel 16K occorre caricare





Tutti assembler appassionatamente

l'uno o l'altro, a seconda delle necessità.

Gli Assembler di questo tipo non prevedono la possibilità di avere in memoria contemporaneamente anche un Disassembler. Le stesse ditte producono peraltro anche Monitor/Disassembler, con caratteristiche dello stesso livello di quelle dei loro Assembler.

C'è da dire che i possessori di 16K non potrebbero in ogni caso avere in memoria i due package, e che proprio ad essi si adattano i prodotti di queste prime categorie, data la loro esigua occupazione di memoria.

Ed eccoci arrivati a prodotti che sfoggiano una certa "eleganza". Si tratta di Assembler con un loro Editor personale, che, se usati dallo Spectrum 48K, possono coesistere in memoria con il loro Disassembler/Monitor e che offrono caratteristiche di tutto rispetto.

Un modello di questa categoria può essere considerato il sistema Editor/Assembler e Monitor/Disassembler della PICTURESQUE.

Un magnifico Editor a 40 colonne, di immediata comprensione, ed un Assembler, che consente una grande libertà, fanno di questo package uno strumento adatto a programmatori evoluti.

Oltre a tutte le normali mnemoniche ed alle pseudo-istruzioni viste per gli altri prodotti, qui c'è la possibilità di effet-

tuare somme e sottrazioni all'interno degli operandi. Per intenderci,

```
ld a,"A"+20
```

dove "A" è un'etichetta di valore precedentemente definito, è perfettamente possibile.

Grande versatilità anche nei modi di definizione delle distanze nei salti relativi: etichette, valori con o senza segno, decimali od esadecimali sono accettati. È inoltre prevista la possibilità di effettuare caricamenti e salvataggi usando il registratore senza uscire dall'Assembler. Il Monitor/Disassembler, studiato per essere usato in unione all'Assembler, non è da meno. Dispone di tutte le caratteristiche fondamentali, con in più una grande versatilità nell'inserimento o cancellazione di blocchi di l/m. Insieme formano un sistema davvero potente.

Salendo ancora troviamo package adatti al solo Spectrum 48K. Si tratta di programmi "seri", adatti a chi sviluppa software complesso ed ha quindi bisogno di particolari caratteristiche.

Diamo un'occhiata allo SPECTRE-MAC-MON, della Oasis Software.

Linee di 254 caratteri con scrolling laterale automatico, possibilità di unire in memoria più file provenienti da nastro, modo "help" per un'informazione rapida sul significato di qualsiasi comando (eh, sì.: iniziano ad essere tanti, per ricordarli tutti) sono le caratteristiche essenziali di questo Editor, che può senz'altro essere definito professionale.

L'Assembler non è da meno: oltre alle mnemoniche, possono essere utilizzate tutte le operazioni algebriche, tutte quelle logiche, parentesi a più livelli, macro con variabili locali. Sono ammesse strutture avanzatissime, come

```
IF..THEN..ELSE
```

```
REPEAT..UNTIL
```

```
WHILE
```

Le caratteristiche da elencare sarebbero ancora moltissime.

Di rilievo la presenza di una copia speciale del package, adatta ad essere trasferita su microdrive (peccato che an-

che in questo caso i file vadano tenuti su cassetta).

Il Monitor/Disassembler è allo stesso livello di qualità. Infatti qui troviamo, oltre a tutte le opzioni classiche, la possibilità di far avanzare l'esecuzione di un programma in l/m di un'operazione alla volta, cosa questa di immenso aiuto nella ricerca di errori.

Superiore sotto certi aspetti, inferiore per altri, è il DEVPAC della HISOFT. Si tratta di un tutt'uno formato da Assembler/Editor e Disassembler/Monitor. L'Assembler dispone in linea di massima delle caratteristiche avanzatissime del Mac-, senza però la sua grande versatilità nella creazione dei macro. D'altra parte qui troviamo una caratteristica unica, la possibilità di assemblare programmi sorgente (cioè composti di mnemoniche) non presenti in memoria, ma su nastro o (udite, udite!) su microdrive. Il caricamento avviene a blocchi di 128 codici, che vengono immediatamente assemblati. Questo sistema è di incalcolabile valore per quei programmi così lunghi da non lasciare spazio per il loro codice assemblato. Tutte le varie (sono molte e diverse) operazioni di caricamento e salvataggi possono essere effettuate con cassetta o con microdrive.

Il Disassembler, ai massimi livelli, dispone della possibilità di esecuzione "passo passo", come il precedente.

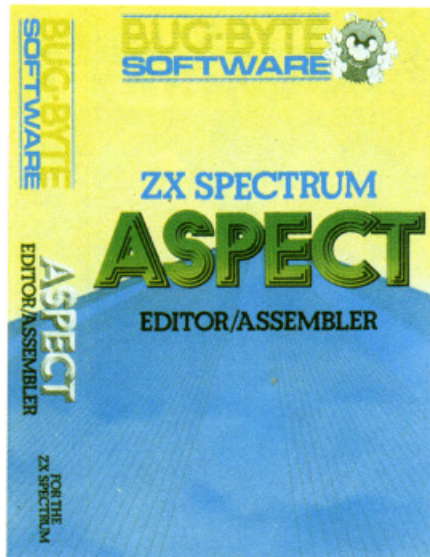
Notevole, nello standard Hisoft, la documentazione.

Per finire, vale la pena di menzionare un ottimo Disassembler, adattissimo ad essere abbinato a quegli Assembler che non ne hanno uno "coordinato". Si tratta del Monitor/Disassembler della Sinclair, prodotto in realtà dalla Crystal Computing. Semplice, completo, versatile, ha anche il pregio di occupare poco spazio in memoria.

Eccoci giunti al termine di questa nostra "escursione" nel campo del software per programmare in linguaggio macchina. Cosa possiamo dire?

Che non è il caso di fare delle classifiche: ogni prodotto ha una sua potenziale utenza. Sarebbe inutile, infatti, che l'utilizzatore a digiuno di Assembly acquistasse un package professionale, spendendo oltretutto un sacco di soldi. Per lui andrà benissimo un più economico e semplice Assembler, cui in un secondo tempo potrà affiancare un buon Disassembler.

Molto più critica è la scelta per chi cerca qualcosa di molto evoluto, anche perché il maggior impegno finanziario consiglia prudenza. Devpac o Mac-Mon? Mah... Intanto, facciamoci una bella partita ad Atic Atac...



Avventura nell'adventure

di **Brian Robb**
 trad. e adatt.
 di **Marcello Spero**

Prima parte

In alternanza con la serie di IF ...THEN dedicata ai linguaggi alternativi, vi presentiamo una serie di articoli che vi spiega come creare un adventure game su ZX81 e Spectrum.

Fra tutti i giochi per computer gli "adventure" sono senz'altro i più difficili da scrivere. È relativamente semplice creare un programma tipo "arcade": l'unico problema è riuscire a far muovere l'"invader" come vogliamo noi. Un programma di avventura deve invece fare i conti con gli esseri umani. Deve apparire intelligente e semplice da seguire, ed inoltre deve poter far fronte a tutte le possibili situazioni create dagli INPUT del frustrato "avventuriero".

Ci sono molti modi per scrivere un programma di avventure, ma l'inizio è uguale per tutti. Il programmatore deve avere una buona idea di avventura ed un piano preciso del suo svolgimento. Si tratta normalmente della parte più dura di tutto il lavoro. Un'ottima fonte di ispirazione è quella "libresca": da Tolkien ad Asimov, non c'è che l'imbarazzo della scelta.

Un'altra fonte, molto spesso usata, è quella televisiva: film e sceneggiati. Persino gli annunci pubblicitari possono suscitare idee notevoli.

In realtà, la maggioranza delle buone idee non emerge da una sola sorgente, ma da un miscuglio di fonti diverse. Prendendo nota di tutte le idee che vi saltano in mente, dopo un po' di tempo potrete accorgervi che un certo numero di queste vanno abbastanza d'accordo per poter creare uno scenario.

Una volta definito lo scenario bisogna tracciare una mappa. Le mappe sono una parte importantissima di ogni av-

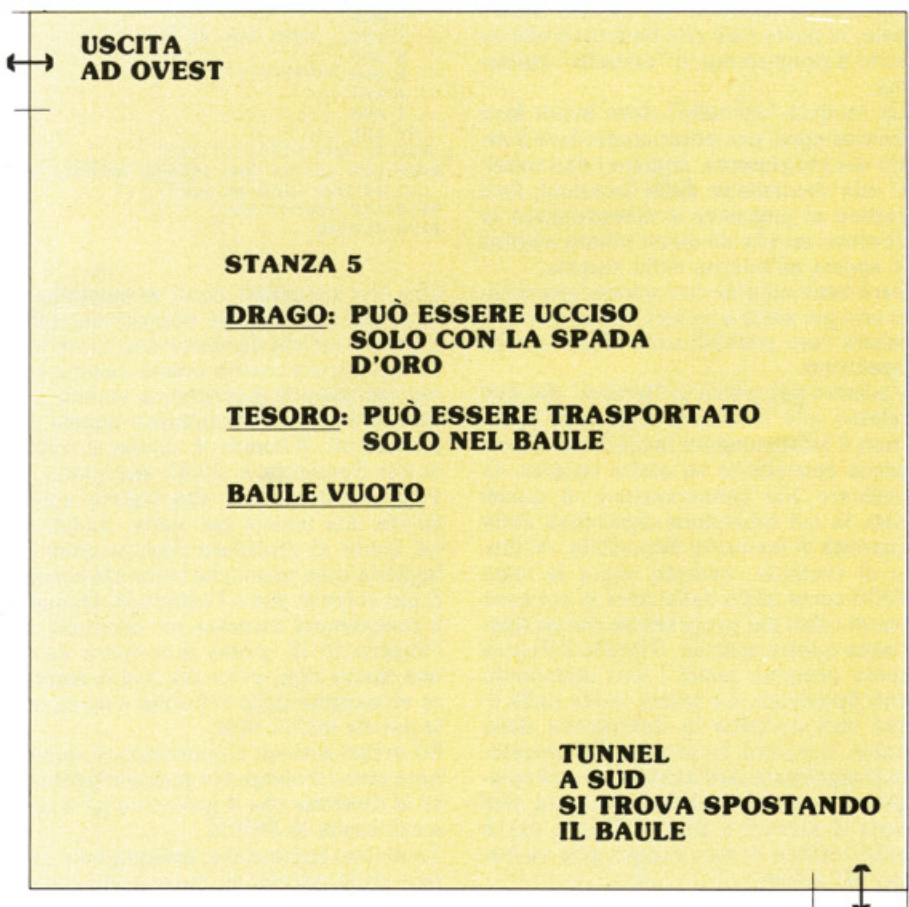
ventura che si rispetti, poiché forniscono un'idea della posizione di ogni "posto" immaginato dalla nostra fantasia. In questo modo ad un'idea astratta viene sostituita una solida e reale rappresentazione.

Crea la nostra mappa, il passo successivo è il suo riempimento con oggetti, mostri, personaggi. Annotate sulla mappa gli oggetti che si possono trovare in ciascuna locazione ed i mostri che potrebbero aggirarsi nell'ombra. Bisog-

nerà inoltre prendere nota di tutte le particolari condizioni necessarie perché un dato mostro possa essere ucciso, o un oggetto raccolto. La figura 1 è un esempio di tutto questo.

Solo quando tutto questo lavoro preliminare sarà stato compiuto potremo metterci di fronte alla tastiera. Certo, è una forte tentazione quella di gettarsi nella stesura di un programma non appena c'è l'idea, ma un comportamento del genere si risolve immancabilmente

fig. 1: esempio di mappa di una stanza



in un pasticcio pieno di errori.

Un programma di avventure può essere costruito come una serie di moduli interconnessi, rendendo più semplice la correzione degli immancabili errori. Le variabili da usare vanno accuratamente elencate e inoltre vanno inizializzate tutte nella prima parte del programma. L'inizializzazione può da sola costituire un modulo. Il successivo potrà contenere le istruzioni del gioco e così via.

Nel listato 1 vedete un tipico modulo di inizializzazione.

Listato 1

```

1 REM *****
2 REM
3 REM   INIZIALIZZAZIONE
4 REM   VARIABILI
5 REM
6 REM *****
7 REM
8 REM
10 LET s=0: REM forza
20 LET qi=100: REM intelligenza
30 LET m=0: REM poteri magici
40 LET ts=0: REM tesoro posseduto
50 LET a$="Cosa vuoi che faccia?":
   REM tipica domanda
60 LET l=0: REM numero locazione
70 LET n=0: REM oggetti trasportati

```

Abbiamo detto come il modo più semplice di costruire un'avventura sia scrivere modularmente, così che i vari "pezzi" possano successivamente venire uniti insieme a formare un'avventura completa. In questo articolo l'attenzione sarà concentrata sul modulo principale, la parte cioè che ha il controllo su tutto il programma: il "cervello" insomma.

Un modulo "cervello" deve saper fare diverse cose, per poter controllare il resto del programma: leggere i dati relativi alla descrizione delle locazioni, farli vedere al giocatore e comprendere le direttive inviate da quest'ultimo mentre lo sposta all'interno della mappa.

Sarà bene considerare queste operazioni una per volta e vedere come possano essere rese possibili sullo ZX81 e sullo Spectrum.

Vediamo per prima la "lettura" dei dati relativi alle locazioni. Con uno Spectrum è relativamente facile farlo. Come prima operazione va scelta un'area da dedicare alla conservazione di questi dati, la cui estensione dipenderà dalla quantità di memoria disponibile. Al titolo di esempio, vedrete usare la linea 9000 come inizio della zona in cui sono tenuti i dati nel programma che accompagna questo articolo. Il listato 2 illustra come vengono tenuti i dati utilizzando uno Spectrum. La prima parte della linea dati contiene la descrizione della prima locazione incontrata dal giocatore (il significato dell'asterisco verrà spiegato in seguito) ed è seguita da una serie di numeri e trattini. Il loro uso è nella routine di movimento, che vedremo più avanti.

Listato 2

```

1 REM *****
2 REM
3 REM   DATI PER SPECTRUM
4 REM
5 REM *****
6 REM
7 REM
9000 DATA "In una stanza buia*si vede
una luce ad est","03_02_"
9010 REM altri dati

```

Se è facile per lo Spectrum accedere ai dati usando l'istruzione READ, con lo ZX81 le cose vanno diversamente, mancando questa macchina delle istruzioni DATA e READ. Si rende perciò necessario un sistema alternativo.

Il listato 3 illustra uno dei sistemi che possono essere usati in luogo di READ e DATA. Lo scopo della linea 90 è quello di definire, all'inizio del gioco, il numero della locazione occupata dal giocatore (cioè la variabile l) come 1. La linea 100 è quella principale. A seconda del valore di l, il programma viene indirizzato ad una differente sezione di dati, in cui l\$ è la descrizione della locazione e d\$ l'insieme delle informazioni relative alla direzione, quest'ultima da usarsi nella routine di movimento.

Listato 3

```

1 REM *****
2 REM
3 REM   DATI PER ZX 81
4 REM
5 REM *****
6 REM
7 REM
90 LET L=1
100 GO SUB 8990+(L*10)
9000 LET L$="IN UNA STANZA BUIA*SI VE
DE UNA LUCE AD EST"
9010 LET D$="03-02-"
9020 RETURN

```

Ora che sappiamo come immagazzinare i dati relativi alla descrizione delle locazioni ed alla direzione (vedremo più avanti come possono essere usati) e come recuperarli, passiamo a vedere come sia possibile visualizzare queste informazioni. Il listato 4 illustra il modo in cui Spectrum e ZX81 stampano le varie locazioni. La descrizione viene scritta una lettera per volta, dando in tal modo al giocatore l'opportunità di leggerla man mano che viene stampata. E qui entra in gioco l'asterisco. Quando il programma incontra un asterisco all'interno di l\$, passa a scrivere su di una nuova riga, evitando così di arrivare al margine dello schermo e spezzare le parole malamente.

Ed eccoci arrivati al momento di esaminare come il computer può comprendere le direttive che il giocatore gli invia, sotto forma di INPUT.

Tanto per cominciare, bisogna fare una lista delle parole, in gran parte verbi,

Listato 4

```

1 REM *****
2 REM
3 REM   STAMPA DATI
4 REM
5 REM *****
6 REM
7 REM
150 CLS
200 PRINT "SEI ";
210 FOR N=1 TO LEN L$
220 IF L$(N)="*" THEN PRINT
230 IF L$(N)="*" THEN LET N=N+1
240 PRINT L$(N);
250 NEXT N
260 IF D$=" STOP " THEN GO TO 7000
270 REM (D$=" STOP " quando il
giocatore viene ucciso)

```

che potranno essere usati nel gioco. I più comuni sono AVANZA.A; PRENDI; LASCIA; INVENTARIO e RIDESCRIVI. In una prossima puntata considereremo alcune parole più particolari. Ovviamente, tutte le parole dovranno essere definite all'inizio del gioco. Il listato 5 illustra come questo possa essere fatto sullo Spectrum, mentre il listato 6 si occupa dello ZX 81. I modi di accesso a questi dati saranno oggetto di un prossimo articolo. I dati sono espressi in forma di parole, contenute in v\$, ciascuna seguita da un numero, contenuto in v, che rappresenta la posizione all'interno del programma della subroutine relati-

Listato 5

```

1 REM *****
2 REM
3 REM   ELENCO PAROLE SPECTRUM
4 REM
5 REM *****
6 REM
7 REM
8000 DATA "AVANZA.A",1000,"PRENDI",20
00,"LASCIA",3000,"INVENTARIO",40
00,"RIDESCRIVI",5000, (e così vi
a)

```

Listato 6

```

1 REM *****
2 REM
3 REM   ELENCO PAROLE ZX 81
4 REM
5 REM *****
6 REM
7 REM
8000 DIM V$(M,N)
8010 DIM V(M)
8020 LET V$(1)="AVANZA.A"
8025 LET V(1)=1000
8030 LET V$(2)="PRENDI"
8035 LET V(2)=2000
8040 REM E COSI' VIA

```

va. Delle varie subroutine, e del modo per accedervi, parleremo la prossima volta. Già questa volta, comunque, prendiamo in considerazione la parola AVANZA.A, che consente il movimento, e la relativa subroutine, quella di movimento, appunto.

Prima di passare ad una qualsiasi subroutine il computer deve però comprendere il significato degli INPUT che

provengono dall'esterno. Questi hanno normalmente la forma di un verbo seguito da un sostantivo con preposizione, tipo "AVANZA.A EST" (la funzione del punto sarà chiarita in seguito).

Il listato 7 illustra come il computer possa capire l'italiano. Il giocatore viene interrogato con la frase: "Cosa vuoi che faccia?", e la risposta viene conservata in z\$. Le linee da 660 a 730 compiono un'operazione detta "slicing", cioè suddivisione di una stringa, il che vuol dire che z\$, cioè in questo caso "AVANZA.A EST" è spezzata in due parole, ciascuna inserita in una stringa diversa; x\$ diventa "AVANZA.A", e y\$ "EST". Ma vediamo meglio questa sezione del programma, chiarendo la funzione di ciascuna linea.

Listato 7

```

1 REM *****
2 REM
3 REM separazione stringhe
4 REM
5 REM *****
6 REM
7 REM
600 PRINT "COSA VUOI CHE FACCIA?"
605 LET Z$=" "; LET X$=" "; LET Y$=" "
610 LET F$=" "
620 PRINT
630 INPUT Z$
640 PRINT " ";Z$
650 PRINT
660 LET Z$=Z$+F$
670 LET Z=0
680 FOR I=1 TO LEN Z$
690 IF Z$(I)=" " AND Z=0 THEN LET X$
   =Z$(1 TO I-1)
700 IF Z$(I)=" " AND Z=1 THEN LET Y$
   =Z$(LEN X$+2 TO I-1)
720 IF X$<>" " THEN LET Z=1
730 NEXT I
    
```

Iniziamo dalla linea 660, che aggiunge uno spazio al termine di z\$ (più avanti vedremo perché). La linea 670 pone un indicatore, o flag, a zero. Ed eccoci alle linee che compiono il lavoro vero e proprio. Sono contenute in un ciclo FOR...NEXT, che inizia alla 680. La lunghezza di questo ciclo dipende da quella di z\$. La linea 690 inizia con

IF z\$(i)=" "...

A questo punto i è uguale a l e z\$(i) è perciò il primo carattere di z\$, e cioè "A". Quindi z\$(i) non è uguale ad uno spazio, ed il programma proseguirà con le linee 700,710 e 720.

La stringa x\$ è stata definita in linea 605 con

LET x\$=" "

cioè con una stringa vuota. Questo ci porta alla linea 730, NEXT I, e quindi di nuovo all'inizio del ciclo. Stavolta, però, i sarà uguale a 2, e quindi sarà z\$(2) ad essere analizzata, cioè la lettera "V". Questo processo prosegue finché z\$(i) non risulta uguale ad uno spazio, cosa che nel nostro esempio accade per i=9, in corrispondenza dello spazio fra "AVANZA.A" e "EST". Questa volta il computer eseguirà fino in fondo la

Listato 8

```

1 REM *****
2 REM
3 REM ROUTINE DI MOVIMENTO
4 REM
5 REM *****
6 REM
7 REM
900 LET L=L
910 IF Y$(1)="N" AND D$( TO 2)<>" _
   " THEN LET L=VAL D$( TO 2)
920 IF Y$(1)="S" AND D$(3 TO 4)<>" "
   " THEN LET L=VAL D$(3 TO 4)
930 IF Y$(1)="E" AND D$(5 TO 6)<>" "
   " THEN LET L=VAL D$(5 TO 6)
940 IF Y$(1)="O" AND D$(7 TO 8)<>" "
   " THEN LET L=VAL D$(7 TO 8)
990 IF L=L2 THEN PRINT "NON E' POSS
   IBILE ANDARE DI LI'"
995 GO TO 100
    
```

linea 690

...AND z=0..

A z è stato dato il valore 0 in linea 670, e da allora in poi è restata immutata. Il risultato dell'operazione logica sarà quindi vero. Z è a zero per indicare che x\$ (che deve contenere la prima parola) è ancora vuota. Ed eccoci finalmente all'ultima parte della linea 690

...THEN LET x\$=z\$(1 TO i-1)

X\$ è tuttora vuoto, e può perciò essere riempito con z\$(1 TO i-1). "AVANZA.A EST" è il contenuto di z\$, ed i è uguale a 9. Quindi z\$(1 TO i-1) sono le prime otto lettere di z\$: "AVANZA.A".

In questo modo x\$ diventa la prima parola della frase. A questo punto il computer raggiunge la linea 700 e, poiché le condizioni di z\$(i) e z sono le stesse di prima, salta alla linea 720. Avendo x\$ il valore "AVANZA.A", la linea 720 viene eseguita interamente, e z diventa 1, ad indicare che x\$ è piena.

La linea 730 incrementa l, portandola a 8. Poiché z\$(8) è "A", il computer prosegue il ciclo finché z\$(i) non sia nuovamente uguale ad uno spazio, e cioè per i=13. Si tratta dello spazio aggiunto al termine di z\$, di cui ora si comprende la funzione: segnalare al computer la fine di un'altra parola. La linea 690 è ignorata, perché, pur essendo z\$(i)=" ", z è ora uguale a 1. Anche la linea 700 è ignorata, ed il computer raggiunge così la linea 710, che viene eseguita, poiché z\$(13) è uno spazio e

z è uguale a 1. Il computer segue l'istruzione

..THEN LET y\$=z\$(LEN x\$+2 TO i-1)

Sembra un'espressione complicata, ma esaminiamola insieme. La stringa y\$ è tuttora vuota, cioè come era stata definita in linea 605. Ora è venuto il momento di darle un valore. LEN x\$ è la lunghezza di x\$, cioè 8, che sommato a 2 dà 10. Il valore di i-1, essendo i=13, è 12. Questo rende l'istruzione appena vista equivalente a

LET y\$=z\$(10 TO 12)

Essendo z\$ uguale a "AVANZA.A EST", le lettere da 10 a 13 sono il blocco "EST", cioè la seconda parola.

In questo modo il computer spezza tutti gli INPUT del giocatore, z\$, in due parole separate, x\$ ed y\$. Tutta l'operazione non dura che qualche secondo, sebbene sembri complessa. Il sistema è stato concepito per lo Spectrum, ma è facilmente convertibile per lo ZX81 con la trasformazione della linea 605 in tre linee separate. A parte questa modifica, tutto il resto può essere usato così com'è.

Finora abbiamo visto come il "cervello" legge i dati relativi alle locazioni e li stampa sullo schermo, nonché come sappia spezzare in due parole gli INPUT del giocatore. Nell'ultima parte di questo articolo, visto che ci siamo occupati finora del verbo "AVANZA.A", diamo un'occhiata alla sezione finale del "cervello", e cioè la routine di movimento.

Il listato 8 riporta la routine al completo, adatta sia allo Spectrum che allo ZX81. È qui che entra in gioco d\$, la riga di numeri e trattini. I listati 2 e 3, per esempio, contengono una stringa dati per la direzione formata da

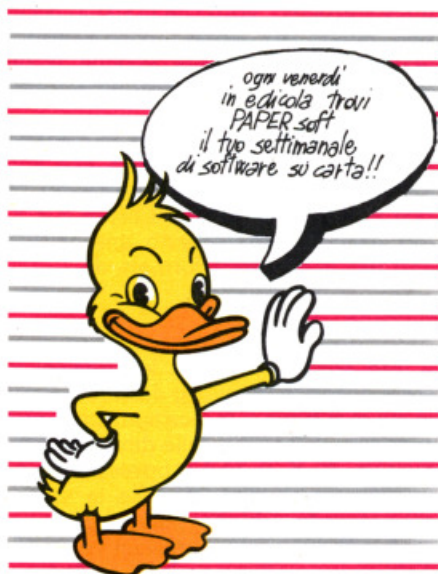
03-02-

Il loro scopo è indicare le direzioni in cui è possibile muoversi dalla locazione in cui ci si trova e le locazioni cui si può accedere. Ciascuna delle direzioni della bussola, N, E, S, O è rappresentata da un numero di due cifre (se la vostra avventura lo richiede, potete aggiungere A e B per le due direzioni verticali). Tale numero altro non è se non la loca-

Nella cassetta allegata a questo numero di Supersinc troverete un programma omaggio: (S) Compattatore di Screen per lo Spectrum 48K realizzato da Andrea De Maria e Emilio Triunfo del Gruppo Utilizzatori Computer Sinclair Napoli.

ERRATA CORRIGE

Nell'articolo "Pascal: il NON GOTO" di Marcello Spero, pubblicato sul numero datato maggio 1985 di SUPERSINC, per un errore tutte le parentesi quadre presenti nel testo sono state sostituite da ampersand (&). Ci scusiamo dell'inconveniente con i lettori.



zione in cui ci si troverà procedendo nella direzione che esso rappresenta. Per fare un esempio, "AVANZA.A NORD" ci porterà nella locazione 03; l diventerà 3 e la descrizione cambierà. Muoversi verso sud non è possibile, come è indicato dal trattino "-", mentre andando ad est ci si troverà nella locazione 02. Anche ovest è una direzione non consentita. La figura 9 riporta le linee di programma usate per decodificare i dati relativi alla direzione.

La linea 900 pone il valore della locazione corrente, cioè l, nella variabile l2. Le linee da 910 in poi si occupano della traduzione della stringa dati. Per renderne chiaro il meccanismo, usiamo l'esempio "AVANZA.A EST".

Dopo che l'INPUT è stato diviso il computer riconosce "AVANZA.A" e salta alla subroutine corrispondente (questa azione è eseguita da un'altra subroutine, che vedremo la prossima volta). Così si giunge alla linea 910, che inizia con

```
IF y$(1)="N"
```

La stringa y\$ contiene la seconda parola, "EST", e quindi y\$(1) è uguale ad "E". Il computer salta le linee 910 e 920 e giunge alla 930. Qui la condizione richiesta,

```
y$(1)="E"
```

è soddisfatta, e quindi l'esecuzione della

linea prosegue. Il seguito è

```
..AND d$(5 TO 6) < > "-"
```

Poiché il numero della locazione è uno, il valore di d\$(5 TO 6) sarà 02, quindi diverso da "-". I valori contenuti in d\$ saranno diversi per ciascuna locazione, a seconda della mappa che avete previsto. Il computer raggiunge quindi l'ultima parte della linea,

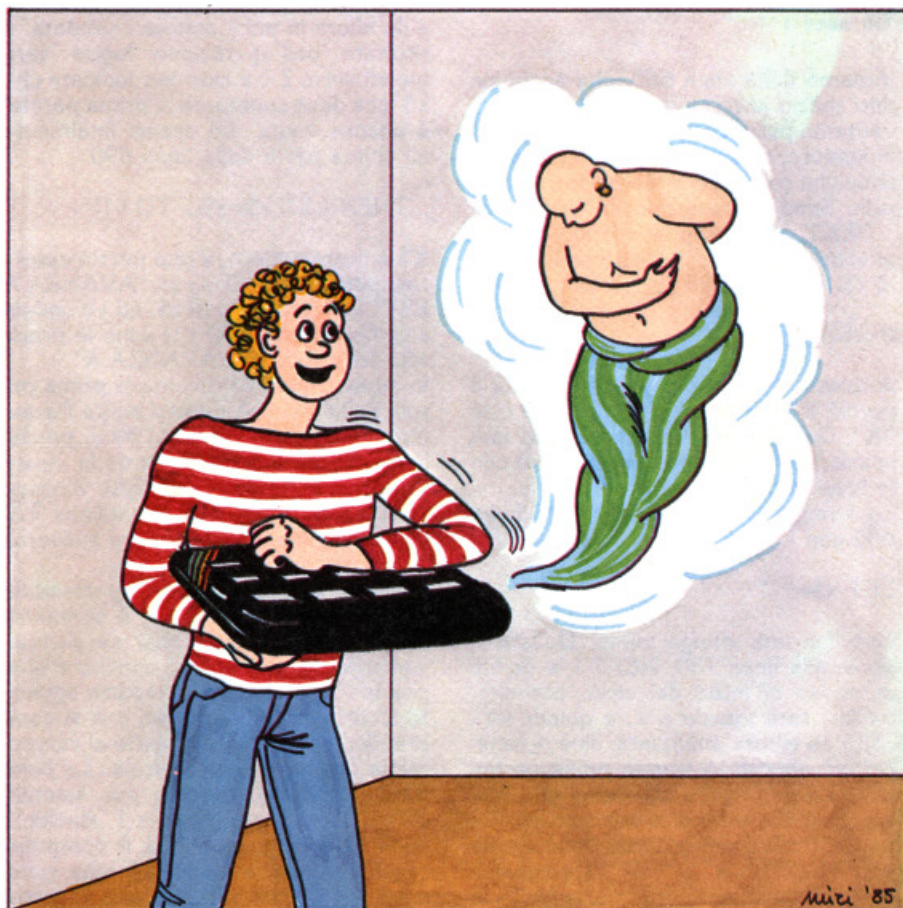
```
..THEN LET l=VAL d$(5 TO 6)
```

che sostituisce all'attuale valore di l quello corrispondente alla locazione che si incontra andando ad est: 02.

La linea 990 ha il compito di verificare se il valore di l è cambiato dal suo precedente, conservato in l2. Se i due valori sono ancora uguali, vuol dire che il giocatore ha tentato di muoversi in una direzione senza uscita.

L'ultima linea ritorna alla linea 100, in cui la descrizione della locazione viene letta e scritta sullo schermo per il giocatore. In questo modo il programma può dirsi di tipo interattivo.

Mettendo insieme tutti i brevi listati che trovate in queste pagine, otterrete un semplice "cervello" di "adventure" e potrete comprenderne il funzionamento. Non si tratta, comunque, di un'avventura completa. Nella prossima puntata vedremo di aggiungere ulteriori subroutine, in modo da ottenere un'avventura completa.



REM:HW

Hardware

Il termostato ben temperato

di **Marcello Spero**

Anche questa volta puntata "tutta soft", dedicata alle applicazioni del sistema di comunicazione con l'esterno da noi realizzato nelle precedenti puntate.

La scorsa volta ci eravamo lasciati con la promessa che avremmo sviluppato un progetto software completo per la regolazione della temperatura. Promessa mantenuta: ecco non uno, ma tre programmi per utilizzare l'insieme interfaccia più convertitore a/d come controllo di temperatura.

Abbiamo visto, nelle puntate scorse, cosa sia un convertitore analogico/digitale (a/d) e come sia possibile realizzarlo in pratica e con poca spesa.

Successivamente ci siamo avventurati nelle problematiche connesse al suo collegamento con un elemento sensibile alla temperatura, esaminando varie soluzioni.

Dopo aver effettuato un sommario collaudo, nella scorsa puntata abbiamo preso in considerazione un sistema più "serio" per utilizzare questo congegno, e cioè una routine di conteggio in linguaggio macchina.

Cosa ha a che fare una routine di conteggio con un convertitore a/d?

La schiera dei lettori affezionati, che ha seguito questa rubrica puntata dopo puntata, conosce bene la risposta a questa domanda. Rivediamo comunque i concetti fondamentali, a beneficio di quanti avessero perso qualche "pezzo" della nostra avventura, o si fossero decisi solo ora a prendere in considerazione l'ipotesi di leggere REM:HW.

È molto semplice; la conversione da quantità di tipo analogico, cioè variabili in modo continuo, a grandezze di tipo digitale, cioè variabili a "scatti" ben definiti che ne costituiscono le unità, può

avvenire in vari modi. Noi abbiamo scelto di trasformare un valore di resistenza, variabile con continuità e quindi analogico, in un valore di tempo. Ma come, dirà qualcuno, anche il tempo è una grandezza variabile in modo continuo! È vero, ma il tempo può essere "sentito" da un computer, mentre una resistenza no. La seconda fase del processo sarà l'approssimazione del valore di questo tempo con una misurazione digitale. Ricordo che le misurazioni digitali di grandezze naturali, e quindi variabili in modo continuo, per quanto precise

possano essere (maggiore precisione si vuole, maggiore sarà il numero di bit da impiegare per esprimere il valore della grandezza) saranno sempre delle approssimazioni del valore originale.

In pratica, il nostro convertitore produce degli intervalli di tempo, proporzionali alle resistenze che gli vengono collegate. Il computer, che come tutti sanno è molto bravo a contare, si incarica di trasformare questo intervallo in un numero, che sarà anche lui proporzionale alla resistenza originale.

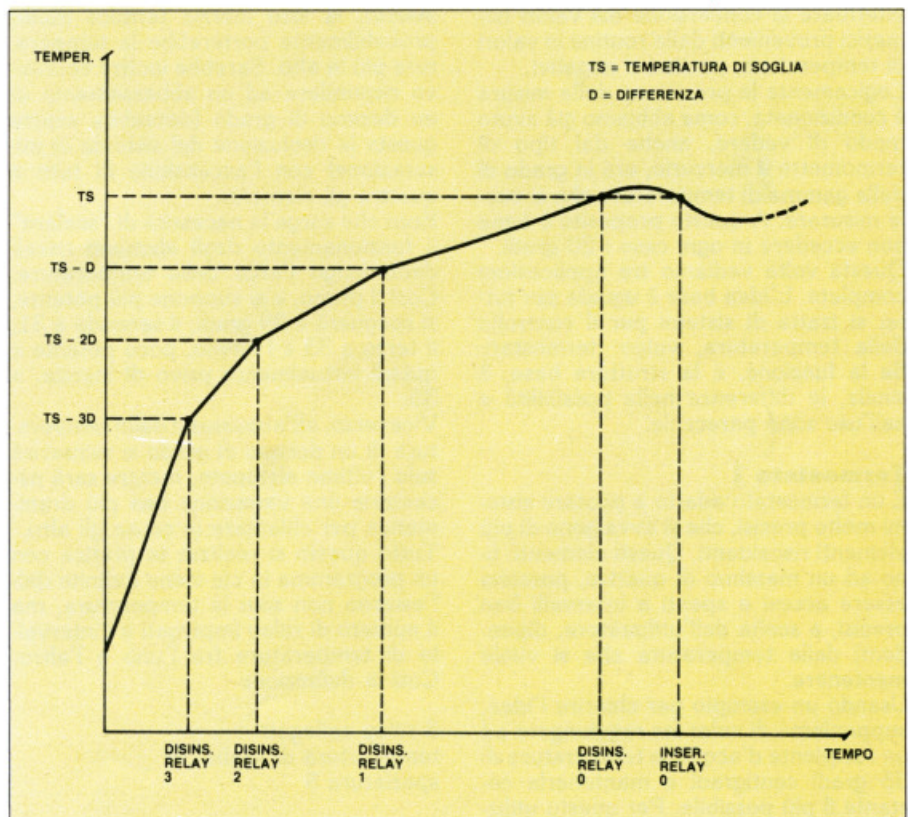


fig. 1: schema di funzionamento del termostato 1

Ecco spiegato il motivo della necessità di una routine di conteggio.

Ovviamente, maggiore è la precisione del conteggio, migliore sarà l'approssimazione della misura. Il BASIC, purtroppo, oltre ad essere relativamente lento, e quindi costringendo ad avere unità di conteggio piuttosto grosse, ha l'intollerabile difetto di non "scorrere" in modo regolare. Esistono infatti, ad intervalli di 20 millisecondi, delle interruzioni necessarie alla lettura della tastiera. La durata di queste interruzioni non è valutabile con precisione, poiché dipende da vari fattori contingenti (modo in cui la macchina si trova, tasto o tasti premuti e così via).

Un buon passo verso la precisione è l'uso del linguaggio macchina. Anche qui, però, occorre fare delle precisazioni. Per motivi legati al modo in cui l'immagine video viene conservata e riprodotta l'accesso da parte della CPU ai primi 16K di RAM può essere momentaneamente inibito. Questo vuol dire che anche il linguaggio macchina, se risiede in questi primi (e per i possessori di Spectrum 16K, unici) 16K, non può dare garanzie di precisione. Questo non vuol dire certo che chi dispone di un 16K non possa provare ad utilizzare le routine di conteggio; la precisione sarà però decisamente inferiore.

A questo punto disponiamo di tutto: elemento termosensibile, convertitore, routine di conteggio. La scorsa volta, anzi, avevamo visto anche un programma di taratura e conversione, che, in base a due dati iniziali ottenuti con un termometro a mercurio era in grado di effettuare la conversione dei valori numerici provenienti dalla routine in valori di temperatura, positivi o negativi. Logicamente la precisione della misura è condizionata, come abbiamo già avuto modo di vedere, anche dal tipo di termometro a mercurio, era in grado di della gamma di temperature che si vuole misurare. A questo proposito è bene non eccedere in ogni caso i 20 gradi. Questa volta vediamo tre applicazioni complete. L'idea base è uguale per tutte; si tratta di sistemi per il controllo della temperatura, vulgo: termostati. Se la funzione, e la struttura base, è simile, le differenze nelle possibilità e nell'uso sono parecchie.

Termostato 1

È un termostato adatto a pilotare sistemi molto precisi, che si avvalgono di più elementi riscaldanti. Questi elementi, fino ad un massimo di quattro, possono essere accesi e spenti a intervalli ben precisi, a scelta dell'utilizzatore, dipendenti dalla temperatura che si vuole mantenere.

Usando un esempio per chiarire l'idea, supponiamo di voler far raggiungere ad un recipiente d'acqua la temperatura di 35 gradi centigradi e mantenerla costante il più possibile. Per questo utilizziamo quattro elementi riscaldanti. Se facessimo arrivare l'acqua fino a 35

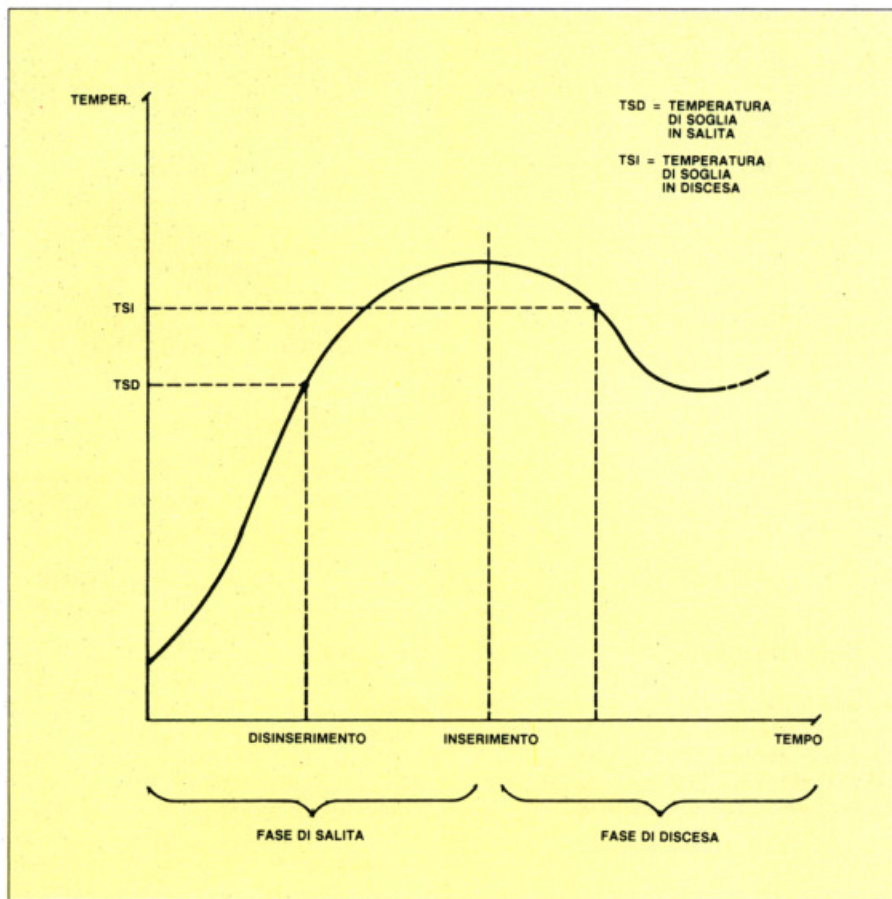


fig. 2: schema di funzionamento del termostato 2

gradi con tutti e quattro gli elementi inseriti, la loro inerzia termica molto probabilmente porterebbe la temperatura più in alto. Sarebbe inoltre eccessivo rispondere ad un abbassamento di un decimo di grado (dovrebbe essere questa la risoluzione del sistema di misurazione) con l'accensione di tutti e quattro gli elementi.

Ecco che sorge la necessità di "scalare" il funzionamento degli elementi riscaldanti lungo l'arco delle temperature. Così il primo, e si suppone più potente, si spegnerà a 29 gradi, il secondo a 31, il terzo a 33 e l'ultimo, poco potente e quindi virtualmente privo di inerzia, a 35.

Viceversa, all'abbassarsi della temperatura di un decimo di grado si reinerà solo l'ultimo elemento, mentre sarà necessaria una variazione ben più drammatica per chiamare in causa gli altri. Tutto questo si realizza in pratica con un programma in cui viene definita dall'esterno non solo la temperatura, ma il numero di relay impiegati e l'intervallo di temperatura fra l'uno e l'altro. Così la definizione

3 relay collegati
temperatura di soglia 38
spaziatura 5

indicherà che i tre relay si disinseriranno rispettivamente a 28, 33 e 38 gradi.

Nel programma ci sono altri due valori che vanno definiti; sono la temperatura di allarme superiore e quella inferiore. Si tratta delle temperature, massima e minima, al di fuori delle quali la situazione può essere considerata di pericolo. Nel caso vengano oltrepassate, normalmente per un guasto degli elementi riscaldanti o di un relay, viene inserito un allarme sonoro.

Si tratta di un termostato pensato per acquari, bagni fotografici e simili.

Termostato 2

Il secondo termostato è adatto invece a grossi impianti di riscaldamento, che normalmente hanno un'inerzia notevole e che non è possibile frazionare in elementi di diversa potenza, come abbiamo fatto prima. Occorre invece cercare di neutralizzare l'inerzia stessa.

Questo è ottenuto definendo due diverse temperature di soglia: una per la "salita" ed una per la "discesa".

In altre parole, volendo mantenere una temperatura di 20 gradi, faremo in modo che l'impianto si disinserisca a 19 gradi, mentre la temperatura sale. Questa, naturalmente, salirà ancora un po' per inerzia. Nel momento in cui inizia la sua discesa entra in vigore un altro limite, che supponiamo sia 21. Dal momento dell'accensione dell'impianto a quello dell'inizio del suo reale riscaldamento passerà del tempo, durante il

quale la temperatura scenderà ulteriormente.

Come si vede, tutto sta nel trovare sperimentalmente la "forbice" di temperature più adatta al proprio impianto.

In questo caso non è prevista la definizione del numero di relay da utilizzare, in quanto abbiamo detto che si tratta di un sistema di pilotaggi per impianti non divisibile. Anche le temperature di allarme qui non esistono, visto che le applicazioni cui è destinato il sistema non dovrebbero essere critiche. Si tratta infatti di un termostato destinato principalmente alla regolazione di temperatura in ambienti domestici.

Termostato 3

Ecco un sistema doppio, adatto cioè a pilotare un elemento riscaldante ed uno raffreddante.

Fissata una temperatura di soglia, al di sotto sarà inserito un relay, al di sopra l'altro.

Ritroviamo qui le due temperature di allarme, visto che le possibili applicazioni includono incubatrici, frigoriferi e celle termostatate.

Vediamo adesso come sono "costruiti" i tre programmi.

Dopo la creazione della schermata iniziale il programma salta alla subroutine di linea 1000, che provvede a caricare l'unico carattere grafico usato (si tratta del cerchietto che indica i gradi centigradi). Si passa quindi alla subroutine

REM:HW

di taratura.

Per prima cosa questa carica il linguaggio macchina della routine di conteggio, di cui vedete l'Assembler nel listato omonimo. Di questa routine c'è qualcosa da dire; infatti non si tratta esattamente di quella presentata la scorsa volta.

Per la precisione, la parte nuova è quella fra le due righe di asterischi. Le modifiche sono volte a far convivere il funzionamento dei relay con quello del convertitore, che utilizzano ambedue l'unità di uscita.

Con il vecchio metodo, infatti, durante il conteggio veniva inviato in uscita prima zero, e quindi 128. In questo modo i bit da 0 a 6 venivano forzati a zero. Questo non ci dava fastidio, nel momento in cui non usavamo i relay, ma diviene intollerabile ora che essi fanno parte integrante del nostro sistema.

La soluzione adottata per "rispettare" lo stato degli altri bit di uscita è quella di porre un bit uguale (cioè contenente lo stato dei vari relay) in una locazione di memoria. Per questo è stata scelta la "solita" variabile di sistema 23728,

normalmente inutilizzata. Al momento del conteggio la routine andrà a leggere questo byte, ne modificherà per i propri usi il solo bit 7 con le operazioni

res 7,a

e

set 7,a

equivalenti alle precedenti

ld a,0

e

ld a,128

e lo invierà in uscita.

Con questo metodo si ottengono due risultati:

- si conserva lo stato degli altri bit, cioè in pratica la posizione dei relay;
- si elimina il bisogno di intervenire con istruzioni IN e OUT dal BASIC per cambiare lo stato di un relay. Basta infatti modificare opportunamente il contenuto dell'indirizzo 23728; la routine di conteggio, che viene richiamata con una frequenza molto alta, penserà ad inviare in uscita la nuova situazione.

Proseguendo nella nostra descrizione, troviamo dalla linea 1040 in poi la subroutine di taratura.

Il suo funzionamento è identico a quello del programma di taratura e conversione visto l'altra volta. Le fasi fondamentali di questa operazione sono:

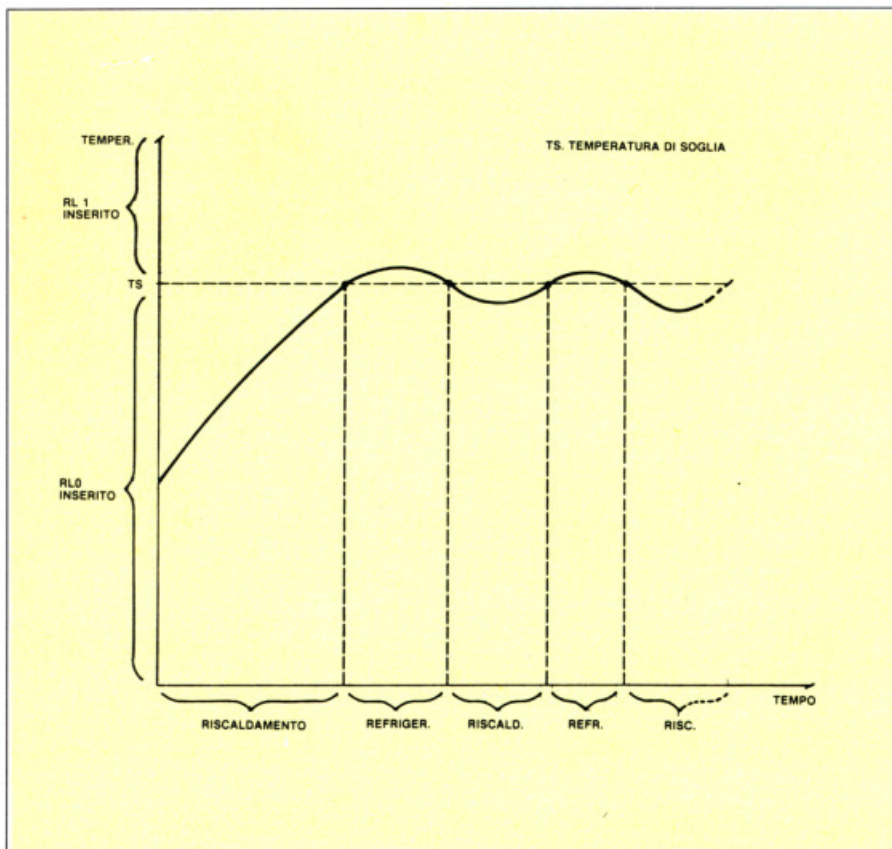
- inserimento della sonda nel recipiente pieno d'acqua ed attesa di una lettura stabile sullo schermo;
- pressione di un tasto per passare oltre ed inserimento del valore di temperatura letto sul termometro a mercurio;
- ripetizione delle stesse operazioni per la seconda temperatura campione, che deve essere maggiore della prima.

Ricordate che le due temperature campione rappresentano i limiti estremi di lavoro del sistema, e più saranno vicine più precise saranno le misure.

Alla taratura segue la routine di lettura, che verrà in seguito richiamata continuamente.

Senza tornare sull'argomento ricordiamo solo che la formula usata per passare dal valore numerico prodotto dal convertitore ad un valore di temperatura è

fig. 3: schema di funzionamento del termostato 3



$$T=T1+(C1-X)/((C1-C2)/(T2-T1))$$

dove T1 e T2 sono le due temperature fornite dall'esterno in corrispondenza dei valori C1 e C2 rilevati dal convertitore.

X è invece il valore letto di volta in volta dal convertitore, che viene trasformato in una misura di temperatura.

Al ritorno dalla subroutine di lettura il programma disporrà di una variabile, temp, contenente il valore della temperatura misurata.

L'uso che di questa viene fatto dipende dal programma.

Nel primo viene calcolata (linee 260-280) la distanza fra questa e la temperatura di soglia, per decidere quanti relay vanno inseriti.

Nel secondo il procedimento è legger-

mente più complesso. La precedente lettura di temperatura si trova immagazzinata nella variabile oldtemp, in modo da consentire, attraverso il suo confronto con temp, di stabilire se la temperatura stia salendo (variabile sale=1) o scendendo (sale=0). A seconda della situazione, verrà presa in considerazione la soglia in salita (tsd) o quella in discesa (tsi).

Il terzo programma, infine, si limita ad inserire un relay se

temp < ts

o l'altro se

temp > ts

La situazione dei relay viene costruita nella variabile poke, che viene quindi

inserita nell'indirizzo 23728.

La parte restante del programma comprende solo le opzioni di ridefinizione dei parametri e di ritaratura.

Non resta che precisare quali sono i relay utilizzati dai vari programmi, informazione che i più attenti avranno già desunto dai listati:

Termostato 1

Lo zero è sempre il primo ad inserirsi e l'ultimo a disinserirsi. A seconda del numero di relay scelto, seguono nell'ordine l'uno, il due ed il tre.

Termostato 2

È utilizzato solo il relay zero.

Termostato 3

Il relay zero pilota l'unità riscaldante, l'uno quella refrigerante.

Termostato 1 versione per ZX Spectrum 48K

```

1 REM
2 REM
3 REM      TERMOSTATO 1
4 REM
5 REM
6 REM
7 REM
10 BORDER 0: PAPER 0: INK 7
20 CLEAR 65000
30 DRAW 255,0
40 DRAW 0,175
50 DRAW -255,0
60 DRAW 0,-175
70 PRINT AT 4,10;"TERMOSTATO 1"
80 PRINT AT 6,12;"taratura"
90 GO SUB 1000
100 PRINT AT 10,13;TAB 25
110 PRINT AT 6,12;TAB 25
120 INPUT "Numero relay?";nrl
130 IF nrl<1 OR nrl>4 THEN GO TO 120
140 PRINT AT 6,6;nrl;" relay colle
gati"
150 INPUT "Temperatura di soglia ";
ts
160 PRINT AT 8,6;"soglia ";ts;"A"
170 INPUT "spaziatura relay?";spr
180 PRINT AT 10,6;"spaziatura ";spr
;"A"
190 INPUT "allarme superiore ";as
200 PRINT AT 12,6;"allarme superiore
";as;"A"
210 INPUT "allarme inferiore ";ai
220 PRINT AT 14,6;"allarme inferiore
";ai;"A"
230 GO SUB 2000
240 IF temp>as OR temp<ai THEN
BEEP 1,40
250 LET poke=128
260 FOR i=0 TO nrl

```

```

270 LET poke=poke+(2↑i AND (temp+i*s
pr<ts))
280 NEXT i
290 POKE 23728,poke
300 PRINT AT 16,6;"temperatura ";tem
p;"A "
310 PRINT AT 18,6;"SPACE per ridefin
ire"
320 PRINT AT 20,6;"ENTER per ritarar
e"
330 LET k$=INKEY$
340 IF k$=CHR$ 32 THEN GO TO 120
350 IF k$=CHR$ 13 THEN GO TO 80
360 GO TO 180
999 STOP
1000 REM
CARICAMENTO UDG
1010 FOR i=USR "a" TO USR "a"+7
1020 READ a: POKE i,a: NEXT i
1030 REM
SUBROUTINE DI TARATURA
1040 FOR i=65001 TO 65025
1050 READ a
1060 POKE i,a
1070 NEXT i
1080 DIM t(2): DIM v(2)
1090 FOR i=1 TO 2
1100 PRINT #1;"un tasto per passare o
ltre"
1110 PRINT AT 10,13;USR 65001;" "
1120 IF INKEY$="" THEN GO TO 1110
1130 INPUT "T";(i);" ";t(i)
1140 LET m=0
1150 PRINT AT 10,6; FLASH 1;"CALCOLO
VALORE MEDIO"
1160 FOR h=1 TO 100
1170 LET m=m+USR 65001
1180 NEXT h
1190 LET v(i)=m/100
1200 PRINT AT 10,6;TAB 26
1210 NEXT i

```

```

1220 LET dt=t(2)-t(1)
1230 LET dv=v(1)-v(2)
1240 LET uv=dv/dt
2000 REM
      ROUTINE DI LETTURA

2010 LET temp=t(1)+(v(1)-USR 65001)/u
      v
2020 LET temp=INT (temp*10)/10
2030 RETURN
2500 REM
      UDG "A"

2510 DATA 96,144,144,96,0,0,0,0
3000 REM
      CODICI L/M

3010 DATA 243,1,0,0,58,176,92,203,191
      ,211,223,203,255,211,223,219,223
      ,183,40,3,3,24,248,251,201

```

Termostato 2 versione per ZX Spectrum 48K

```

1 REM
2 REM
3 REM      _____
4 REM      TERMOSTATO 2
5 REM      _____
6 REM
7 REM
10 BORDER 0: PAPER 0: INK 7
20 CLEAR 65000
30 DRAW 255,0
40 DRAW 0,175
50 DRAW -255,0
60 DRAW 0,-175
70 PRINT AT 4,10;"TERMOSTATO 2"
80 PRINT AT 6,12;"taratura"
90 GO SUB 1000
100 PRINT AT 10,13;TAB 25
110 PRINT AT 6,12;TAB 25
120 INPUT "Temperatura disinseriment
o ";tsd
130 PRINT AT 8,6;"Temperatura disins
erimento ";tsd;"A"
140 INPUT "Temperatura inserimento
";tsi
150 PRINT AT 10,6;"Temperatura inser
imento ";tsi;"A"
160 LET oldtemp=0
170 LET sale=0
180 GO SUB 2000
190 LET poke=128
200 IF temp>oldtemp THEN LET sale=1

210 IF temp<oldtemp THEN LET sale=0

220 LET poke=poke+(((sale=1) AND (te
mp<tsd)) OR ((sale=0) AND (temp<
tsi)))
230 POKE 23728,poke
240 PRINT AT 12,6;"temperatura ";tem
p;"A "

```

```

250 PRINT AT 14,6;("in salita" AND s
ale=1)+("in discesa" AND sale=0)

260 PRINT AT 16,6;"relay ";("dis"
AND poke=128);"inserito"
270 PRINT AT 18,6;"SPACE per ridefin
ire"
280 PRINT AT 20,6;"ENTER per ritarrar
e"
290 LET k$=INKEY$
300 IF k$=CHR$ 32 THEN GO TO 120
310 IF k$=CHR$ 13 THEN GO TO 80
320 GO TO 180
999 STOP
000 REM

```

CARICAMENTO UDG

```

010 FOR i=USR "a" TO USR "a"+7
020 READ a: POKE i,a: NEXT i
030 REM

```

SUBROUTINE DI TARATURA

```

1040 FOR i=65001 TO 65025
1050 READ a
1060 POKE i,a
1070 NEXT i
1080 DIM t(2): DIM v(2)
1090 FOR i=1 TO 2
1100 PRINT #1;"un tasto per passare o
ltre"
1110 PRINT AT 10,13;USR 65001;" "
1120 IF INKEY$="" THEN GO TO 1110
1130 INPUT "T";(i);" ";t(i)
1140 LET m=0
1150 PRINT AT 10,6; FLASH 1;"CALCOLO
VALORE MEDIO"
1160 FOR h=1 TO 100
1170 LET m=m+USR 65001
1180 NEXT h
1190 LET v(i)=m/100
1200 PRINT AT 10,6;TAB 26
1210 NEXT i
1220 LET dt=t(2)-t(1)
1230 LET dv=v(1)-v(2)
1240 LET uv=dv/dt
2000 REM

```

ROUTINE DI LETTURA

```

2010 LET temp=t(1)+(v(1)-USR 65001)/u
      v
2020 LET temp=INT (temp*10)/10
2030 RETURN
2500 REM

```

UDG "A"

```

2510 DATA 96,144,144,96,0,0,0,0
3000 REM

```

CODICI L/M

```

3010 DATA 243,1,0,0,58,176,92,203,191
      ,211,223,203,255,211,223,219,223
      ,183,40,3,3,24,248,251,201

```

Termostato 3
versione per ZX Spectrum 48K

```

1 REM
2 REM
3 REM      TERMOSTATO 3
4 REM
5 REM
6 REM
7 REM
10 BORDER 0: PAPER 0: INK 7
20 CLEAR 65000
30 DRAW 255,0
40 DRAW 0,175
50 DRAW -255,0
60 DRAW 0,-175
70 PRINT AT 4,10;"TERMOSTATO 3"
80 PRINT AT 6,12;"taratura"
90 GO SUB 1000
100 PRINT AT 10,13;TAB 25
110 PRINT AT 6,12;TAB 25
120 INPUT "Temperatura di soglia ";
    ts
130 PRINT AT 8,6;"soglia ";ts;"A"
140 INPUT "allarme superiore ";as
150 PRINT AT 10,6;"allarme superiore
    ";as;"A"
160 INPUT "allarme inferiore ";ai
170 PRINT AT 12,6;"allarme inferiore
    ";ai;"A"
180 GO SUB 2000
190 IF temp>as OR temp<ai THEN
    BEEP 1,40
200 LET poke=128
210 LET poke=poke+(temp<ts)+(2 AND t
    emp>ts)
220 POKE 23728,poke
230 PRINT AT 14,6;"temperatura ";tem
    p;"A "
240 PRINT AT 16,6;"SPACE per ridefin
    ire"
250 PRINT AT 20,6;"ENTER per ritara
    re"
260 LET k$=INKEY$
270 IF k$=CHR$ 32 THEN GO TO 120
280 IF k$=CHR$ 13 THEN GO TO 80
290 GO TO 180
999 STOP
000 REM

```

CARICAMENTO UDG

```

010 FOR i=USR "a" TO USR "a"+7
020 READ a: POKE i,a: NEXT i
030 REM

```

SUBROUTINE DI TARATURA

```

040 FOR i=65001 TO 65025
050 READ a
060 POKE i,a
070 NEXT i
080 DIM t(2): DIM v(2)
090 FOR i=1 TO 2
100 PRINT #1;"un tasto per passare o
    ltre"

```

```

1110 PRINT AT 10,13;USR 65001;" "
1120 IF INKEY$="" THEN GO TO 1110
1130 INPUT "T";(i);" ";t(i)
1140 LET m=0
1150 PRINT AT 10,6; FLASH 1;"CALCOLO
    VALORE MEDIO"
1160 FOR h=1 TO 100
1170 LET m=m+USR 65001
1180 NEXT h
1190 LET v(i)=m/100
1200 PRINT AT 10,6;TAB 26
1210 NEXT i
1220 LET dt=t(2)-t(1)
1230 LET dv=v(1)-v(2)
1240 LET uv=dv/dt
2000 REM

```

ROUTINE DI LETTURA

```

2010 LET temp=t(1)+(v(1)-USR 65001)/u
    v
2020 LET temp=INT (temp*10)/10
2030 RETURN
2500 REM

```

UDG "A"

```

2510 DATA 96,144,144,96,0,0,0,0
3000 REM

```

CODICI L/M

```

3010 DATA 243,1,0,0,58,176,92,203,191
    ,211,223,203,255,211,223,219,223
    ,183,40,3,3,24,248,251,201

```

Assembler della routine di conteggio

```

50 REM
55 REM
60 REM      ASSEMBLER
70 REM      ROUTINE
80 REM      DI CONTEGGIO
85 REM      MODIFICATA
90 REM
100 REM
105 REM
110 REM      org 50001
115 REM
120 REM      di
130 REM      ld bc,00
135 REM      *****
140 REM      ld a,(23728)
145 REM      res 7,a
150 REM      out (223),a
160 REM      set 7,a
170 REM      out (223),a
175 REM      *****
180 REM LOOP;in a,(223)
190 REM      or a
200 REM      jr z,FINE
210 REM      inc bc
220 REM      jr LOOP
240 REM FINE;ei
250 REM      ret

```

RANDOMIZE

L'Assembly assieme

di **Marcello Spero**

Terza parte

Approfondiamo insieme la conoscenza del linguaggio macchina di Spectrum e ZX81

Eccoci ancora una volta a percorrere insieme la strada, qualche volta liscia ed in discesa, più spesso piena di buche e ostacoli, della conoscenza del linguaggio macchina di Spectrum e ZX81.

Ormai siamo a buon punto, e i più coraggiosi si saranno certamente già cimentati nella stesura delle loro prime creazioni. Stiamo procedendo troppo lentamente? Abbiate pazienza: non tutti coloro che ci seguono possiedono la stessa esperienza. Corriamo troppo? Se qualcosa vi è sfuggito, ripercorrete da soli il cammino già fatto, rileggendo le puntate precedenti.

Da lontano il vento porta alcune flebili proteste, che riguardano gli esempi, troppo banali. Era necessario che fossero così, per poter essere compresi da tutti. Fin da questa puntata, comunque, dovrete ricredervi (fermi! Non saltate avanti).

Ci eravamo lasciati, la volta scorsa, con due routine che riassumevano un po' tutto quello che avevamo visto fino ad allora. Né facili né difficili, mancavano comunque di qualsiasi spiegazione. Dopo aver lasciato a voi per un mese intero l'ingrato compito di capire come facessero le dannate routine a fare quello che fanno, vediamo insieme la meccanica e le possibili modifiche.

Per lo ZX81 la routine presentata era

```
ld a, (16396)
ld e,a
ld a, (16397)
ld d,a
ld a,33
ld c,22
ld b,32
ld (de),a
inc de
djnz UNO
inc de
dec c
jr nz, DUE
ret
```

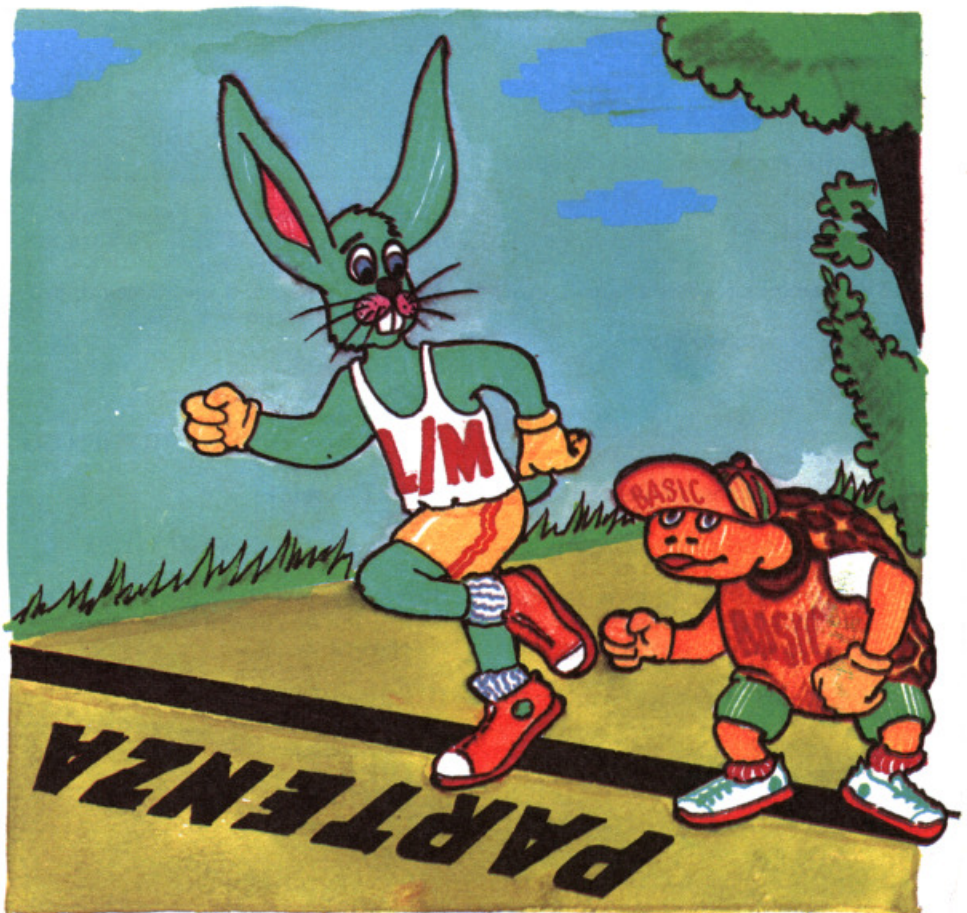
DUE
UNO

Il suo effetto, come tutti gli "zetaottantunisti" avranno constatato, è il riempimento dello schermo con un carattere, il numero 5. Data l'elevatissima velocità, non è possibile seguire visivamente la successione delle operazioni che portano all'effetto finale. Per comprendere il funzionamento della routine dobbiamo perciò guardare com'è costruita. Il primo blocco di operazioni da considerare è

```
ld a, (16396)
ld e,a
ld a, (16397)
ld d,a
```

Per capirne il significato bisogna sapere che nello ZX81 la posizione dell'area di memoria destinata a contenere i caratteri che compongono l'immagine video non è fissa. Per questo esiste una variabile di sistema, D-FILE, all'indirizzo 16396, che contiene il valore del suo primo byte, cioè il suo inizio.

Noi vogliamo riempire tutti i caratteri dell'immagine con dei "5"; siccome anche qui, come in tutte le cose, è bene partire dall'inizio, caricheremo in una coppia di registri il valore del primo byte dell'area video (chiamata Display File in "sinclairese"). Ma perché usare una coppia di registri?



Semplice: perché gli indirizzi di memoria sono numeri da 0 a 65535, e quindi un solo registro non ce la farebbe a contenerne tutti i possibili valori. Per lo stesso motivo la variabile di sistema D-FILE è composta da due byte.

La coppia prescelta è de. In d andrà la metà più significativa del valore (cioè quella superiore, in cui ogni unità vale 256), in e quella meno significativa (cioè quella inferiore). Poiché in memoria vale la regola generale (vi ricordate?) che il primo byte di una coppia sia quello meno significativo ed il secondo quello più significativo, dovremo prendere il contenuto di 16396 e copiarlo in e; il contenuto di 16397 andrà invece in d. L'unico modo che conosciamo per trasferire il contenuto di una locazione di memoria in un registro è

ld a,(% < indirizzo > %)

per cui siamo costretti ad usare

```
ld a,(% < indirizzo 1 > %)
ld e,a
ld a,(% < indirizzo 2 > %)
ld d,a
```

Potremmo agire in modo diverso e più rapido?
L'operazione

ld (% < coppia > %),(% < indirizzo > %)

esegue da sola tutto il procedimento:

- carica con il contenuto dell'indirizzo il secondo registro della coppia;
- incrementa l'indirizzo;
- carica con il contenuto del nuovo indirizzo il primo registro della coppia.

Quindi, nel nostro caso, la sola operazione

ld de,(16396)

può validamente sostituire tutto il primo blocco.

Passiamo al secondo blocco:

```
ld a,33
ld b,22
ld c,32
```

Si tratta del caricamento dei vari valori:

- in a il codice del carattere scelto, cioè il numero 5 (chr\$ 33);
- in b il numero di righe da riempire (22, cioè tutto lo schermo);
- in c il numero di colonne di ogni riga (32).

Il terzo blocco, cioè

```
DUE ld b,32
UNO ld (de),a
inc de
dynz UNO
inc de
dec c
jr nz, DUE
```

Come caricare una routine in linguaggio macchina

La prima risposta, fin troppo ovvia, è quella di procurarvi un programma assembler e di usarlo; se non lo possedete o preferite lavorare col linguaggio macchina, restando nel più sicuro e comodo ambiente BASIC, usate i seguenti programmi.

Spectrum

```
10 CLEAR 29999
20 LET X=30000
30 INPUT a
40 POKE x,a
50 PRINT x, PEEK x
60 LET x=x+1
70 GO TO 30
```

Il valore a cui viene abbassata RAMTOP, con CLEAR, è adatto per tutti gli Spectrum.

ZX81

Per lo ZX81 la questione è leggermente più complicata: innanzitutto occorre modificare RAMTOP, con

```
10 POKE 16388,93
20 POKE 16389,191
```

dopo di che date NEW. Ora potete utilizzare un programma simile a quello per lo Spectrum.

```
10 LET X=24000
20 INPUT A
30 POKE X,A
40 PRINT X,PEEK X
50 LET X=X+1
60 GO TO 20
```

Per attivare le routine introdotte dovrete dare, a seconda dei casi, RANDOMIZE USR (o PRINT USR) 30000 per lo Spectrum, e RAND USR (o PRINT USR) 24000 per lo ZX81.

compie materialmente l'operazione di riempimento dei 32*22=704 caratteri del video, uno per uno.

Per comprenderne il meccanismo, proviamo a riscriverlo in BASIC:

```
10 LET b=32
20 POKE de,a
30 LET de=de+1
40 LET b=b-1:IF b < > 0 THEN GO TO 20
50 LET de=de+1
60 LET c=c-1
70 IF c < > 0 THEN GO TO 30
```

che corrisponde "letteralmente" all'Assembly, ma può essere trasformato, più sinteticamente, in

```
10 FOR c=1 TO 22
20 FOR b=1 TO 32
30 POKE de,a
40 LET de=de+1
50 NEXT b
60 LET de=de+1
70 NEXT c
```

in cui sono più evidenti i due cicli, inseri-

ti l'uno dentro l'altro (si dice in questo caso che sono "nidificati"). Il ciclo più interno "percorre" ciascuna riga, mentre il più esterno controlla il numero di righe totali, cioè in pratica le colonne. Ad ogni giro del ciclo più interno il valore di a viene inserito all'indirizzo opportuno, e quindi de viene incrementato. Al termine di ogni riga de viene incrementato una volta più del solito, per scavalcare il carattere "newline", che chiude la riga.

Tornando all'Assembly, notate come il ciclo più interno, utilizzando il registro b come variabile di controllo, possa essere realizzato con dynz.

Quello più esterno, invece, dovendo usare c, deve per forza essere di tipo esteso, cioè

```
dec < registro >
jr < condizione >
```

Le possibili modifiche sono infinite. La più immediata è la sostituzione del carattere "5" con un altro qualsiasi; per farlo basta sostituire, nell'operazione

ld a,33

il codice del numero 5 (33, appunto) con quello del carattere desiderato, desumibile dall'appendice del manuale. Altra cosa piuttosto semplice è la limitazione del numero di righe da riempire, diminuendo il valore iniziale di c, cioè quello che viene impostato dall'operazione

ld c,22

Più complesso sarebbe invece ridurre il numero di caratteri per linea, o partire da un determinato carattere che non sia il primo. Ma di questo ci occuperemo un'altra volta.

Cedendo alle proteste degli "spectromani" passiamo ad esaminare la seconda routine, quella per Spectrum, appunto. Si trattava di una routine di "scroll", cioè di spostamento, laterale di tutti i caratteri dello schermo verso destra di una posizione. Rivediamola:

```

ld de,16384
ld c,192
DUE ld b,32
ld h,0
UNO ld a,(de)
ld l,a
ld a,h
ld (de),a
ld h,l
inc de
dync UNO
dec c
jr nz,DUE
ret

```

Qui si tratta di spostare materialmente verso destra di un byte il contenuto di tutti i byte dell'area video. In figura 2 vedete l'architettura di quest'area nello Spectrum. Si tratta di una struttura certo più complessa di quella dello ZX81, ma che offre anche numerosi vantaggi. Prima di tutto, la sua posizione in memoria è fissa, e parte sempre dall'indirizzo 16384. Poi, non esistono caratteri "newline" o "enter" al termine delle righe: all'ultimo carattere di ogni riga segue subito il primo della riga successiva. L'unico (!) grosso problema riguarda la disposizione dei vari byte che formano i caratteri; ogni carattere è infatti composto da ben otto byte, che sullo schermo televisivo si dispongono uno sull'altro. In memoria, però, la loro posizione segue criteri più complessi, legati alla meccanica della composizione dell'immagine televisiva. Quello che ci interessa, comunque, è che questi byte sono sparpagliati solo in senso verticale. In altre parole, i byte relativi ai caratteri della prima colonna (0,0;1,0;2,0;3,0;...) saranno sempre nella prima colonna, anche se le righe non verranno rispettate. Questo ci consente di agire come se tutto fosse in ordine. Perché? Semplice: a noi interessa che tutto quello che è nella prima colonna passi nella seconda, quello che è nella seconda passi nella terza e così via, e per fare questo non

ci sono problemi.

Ben altra sarebbe la complessità di uno scrolling verticale, ma per fortuna a questo ha già pensato la Sinclair.

In sostanza, il nostro problema è quello di spostare a destra 192 righe (24*8) di 32 caratteri ciascuna. Come vedete, per semplicità agiamo anche sulle ultime due righe, quelle cioè destinate ai comandi.

Suddividiamo ora la routine in blocchi funzionali. Il primo è

```

ld de,16384
ld c,192
ld b,32

```

che inizializza i vari registri:

- de che conterrà l'indirizzo dell'area video su cui si opera;

- c che conterrà le righe;

- b che conterrà le colonne

Il secondo blocco è

```

DUE ld b,32
ld h,0
UNO ld a,(de)
ld l,a
ld a,h
ld (de),a
ld h,l
inc de
dync UNO
dec c
jr nz,DUE

```

che compie tutto il lavoro di trasferimento. Come abbiamo fatto prima, scriviamone l'equivalente in BASIC

```

10 LET b=32
20 LET h=0
30 POKE a,de
40 LET l=a
50 LET a=h
60 POKE de,a
70 LET h=l
80 LET de=de+1
90 LET b=b-1:IF b < > 0 THEN GO TO 30
100 LET c=c-1
110 IF c < > 0 THEN GO TO 10

```

riducibile alla più compatta e chiara forma

```

10 FOR c=1 TO 192
20 FOR b=1 TO 32
30 LET h=0
40 POKE a,de

```

```

50 LET l=a
60 LET a=h
70 POKE de,a
80 LET h=l
90 LET de=de+1
100 NEXT b
110 NEXT c

```

in cui si riconoscono i due cicli, l'uno per le colonne, più interno, comandato da b, l'altro per le righe, più esterno, comandato da c. Il procedimento di spostamento, apparentemente complesso, è riassumibile in:

- copia il contenuto dell'indirizzo corrente;
- mettilo in l;
- metti nell'indirizzo corrente il contenuto dell'indirizzo precedente (e quindi più a sinistra), contenuto in h. Per tutti i byte della prima colonna questo contenuto non esiste, e verrà perciò inserito uno spazio (cioè un byte a zero);
- sposta il contenuto recuperato da l ad h, in modo che sia pronto per essere trasferito al prossimo byte;
- incrementa l'indirizzo di lavoro (cioè de).

Il contenuto di tutti i byte dell'ultima colonna va perso. Infatti il registro h, all'uscita dal ciclo più interno, viene sovrascritto con uno spazio. In questo modo si evitano indesiderati "ritorni" di immagine sulle righe successive. La figura 3 schematizza il movimento nel suo complesso.

Quali sono le modifiche possibili per snellire la routine?

Visto che lo spostamento deve avvenire verso destra, si potrebbe iniziare dalla fine dell'area video, ed in questo modo eliminare la necessità di conservare in h ed l il contenuto del byte che viene modificato. Vediamo perché:

- l'ultimo byte dell'area viene riempito con il contenuto del penultimo (il suo contenuto va perso, come per tutti i byte dell'ultima colonna);

- il penultimo byte viene riempito con il contenuto del terzultimo;

- arrivati al byte in prima colonna lo si riempie con uno spazio e si riprende con la riga superiore.

Ecco come appare la routine:

```

ld de,22527
ld c,192
UNO ld b,31
DUE dec de
ld a,(de)

```

RANDOMIZE

```
inc de
ld (de),a
dec de
djnz DUE
ld a,0
ld (de),a
dec de
dec c
jr nz, UNO
ret
```

Seppure interessante, questa modifica non porta alcun vantaggio in termini di spazio.

Più interessante è invece la sostituzione della coppia de con hl. Questa coppia è molto particolare, e svolge più o meno la stessa funzione del registro a, nel campo dei numeri a due byte. Se, infatti, l'operazione

```
ld <registro> ,(% <coppia> %)
```

non è consentita, essendo permessa solo

```
ld a,(% <coppia> %)
```

l'operazione

```
ld <registro> ,(hl)
```

è perfettamente lecita. Le operazioni possibili solo ad hl sono molte, e le vedremo poco per volta. Per il momento ci basta sapere che possiamo riscrivere la nostra routine così:

```
ld hl,16384
ld c,192
DUE ld b,32
ld h,0
UNO ld l,(hl)
ld (hl),h
ld h,l
inc hl
djnz UNO
dec c
jr nz,DUE
ret
```

guadagnando in brevità e chiarezza. Dopo aver scandagliato in tutti i modi possibili le nostre due routine passiamo a qualcosa di nuovo.

Questa volta non si tratta di nuove operazioni, ma di metodi più avanzati per lo scambio di dati fra un programma BASIC ed una routine in linguaggio macchina.

Fin dalla prima puntata sappiamo come far eseguire una routine in linguaggio macchina: con

```
RANDOMIZE USR <indirizzo>
```

Sappiamo anche che è possibile ricevere dalla routine stessa un valore, che dovrà essere contenuto nella coppia hl per lo ZX81 o nella coppia bc per lo Spectrum. Questo valore verrà attribuito, al momento del ritorno al BASIC, alla funzione USR. Dando quindi

```
PRINT USR <indirizzo>
```

ne otterremo la stampa, mentre con

```
LET a=USR <indirizzo>
```

lo assegneremo alla variabile a, per esempio.

In questo modo, però, non è possibile passare dati dal BASIC alla routine, all'inizio della sua esecuzione.

Per questo esistono diverse tecniche. La più semplice è la costituzione di un'area dati, al di sopra della RAMTOP, nella quale disporre ordinatamente i vari valori necessari alla routine con delle POKE.

Supponiamo di aver deciso di collocare i nostri dati a partire dall'indirizzo 30001. Le operazioni da fare saranno:

– Portare la RAMPTOP al di sotto di 30001, se non lo è già, con

```
CLEAR 30000
```

per lo Spectrum.

– Introdurre i vari dati (10 nell'esempio che segue) con

```
FOR i=30001 TO 30010
INPUT a
POKE i,a
NEXT i
```

o

```
FOR i=30001 TO 30010
POKE i,a(i-30000)
NEXT i
```

a seconda che si tratti di dati provenienti dall'esterno o già contenuti in un vettore. Questo se si tratta di dati i cui valori sono compresi fra 0 e 255, e quindi contenibili in un solo byte. Altrimenti occorrerà

```
FOR i=30001 TO 30020 STEP 2
INPUT a
POKE i,a-256*INT(a/256)
POKE i+1,INT(a/256)
NEXT i
```

o

```
FOR i=30001 TO 30020
POKE i,a(i-30000)
-256*INT(a(i-30000)/256)
POKE i+1,INT(a(i-30000)/256)
NEXT i
```

La routine potrà recuperare questi dati al momento opportuno con operazioni tipo

```
ld <registro> ,(30001)
ld <registro> ,(30002)
```

e così via.

Di questo sistema di trasferimento dati vediamo un esempio per lo ZX81.

Si tratta di una routine che stampa un determinato carattere in una certa posizione dello schermo. Carattere e posizione sono stabiliti dall'utilizzatore.

Tutto il discorso fatto per lo Spectrum vale anche per il "fratellino minore", con la differenza che qui non possiamo certo partire dall'indirizzo 30001, che non esiste. Prendiamo come inizio dell'area dati l'indirizzo 24101, supponendo di porre la routine all'indirizzo 24001, come abbiamo sempre fatto con questa macchina.

Decidiamo di usare questa mappa di memoria per la nostra area:

indirizzo	contenuto
24101	codice carattere
24102	numero riga
24103	numero colonna

Introdurremo i dati con il programmino BASIC

```
10 FOR i=24101 TO 24103
20 INPUT a
30 POKE i,a
40 NEXT i
```

dopo aver introdotto la routine. L'abbassamento della RAMTOP per proteggere quest'ultima servirà anche a proteggere l'area dati.

Ed ecco la routine:

```
ld hl,(16396)
ld a,(24102)
cp 0
jr z, LINEA0
ld b,a
CICLO add hl,32
djnz CICLO
LINEA0 ld a,(24103)
ld e,a
ld d,0
add hl,de
ld a,(24101)
ld (hl),a
ret
```

Anche su questa routine c'è qualcosa da dire.

Prima di tutto, il metodo usato per trovare l'indirizzo del carattere, partendo dalle sue coordinate. Si tratta dell'equivalente della formula

I=BASE+R*32+C

dove I è l'indirizzo finale, BASE è l'indirizzo di inizio del D-File, R è il numero della riga e C quello della colonna. La moltiplicazione viene realizzata in linguaggio macchina (in questo caso) con addizioni ripetute. Quindi, in sostanza, l'equivalente di

```
LET I=PEEK 16396+256*PEEK
16397
FOR k=1 TO R
LET I=I+32
NEXT k
LET I=I+C
```

in BASIC. Ma osserviamo più da vicino la routine. Anzi, descriviamone le sue operazioni una ad una, visto che le cose nuove sono più di una.

ld hl,(16396)

– Caricamento dell'indirizzo di inizio del D-File nella coppia hl, che alla fine del calcolo conterrà l'indirizzo finale. Date le dimensioni del valore è necessaria una coppia. Il caricamento avviene dalla variabile di sistema 16396, che contiene appunto tale indirizzo. Come abbiamo già avuto occasione di vedere, questa operazione effettua da sola il caricamento da due locazioni di memoria consecutive.

ld a,(24102)

– Caricamento del numero di riga (vedi mappa dell'area dati).

Le due operazioni che seguono servono a verificare se il numero di riga per caso non è 0, ed in tal caso evitare il procedimento di moltiplicazione. Per trovare un indirizzo in riga 0, infatti, basta sommare alla base il numero di colonna.

cp 0

– Confronto del contenuto di a con 0. L'operazione

cp N

(compare), che vediamo qui per la prima volta, è sostanzialmente identica alla sottrazione

sub N

con l'importante differenza che il risultato viene scartato, anziché essere posto in a, che pertanto conserva il suo vecchio valore. Ma allora, a cosa serve? A modificare i flag secondo il risultato della sottrazione, cioè in pratica a confrontare i due numeri. Nel nostro caso, se a=0, avremo

0-0=0

risultato 0, e quindi flag di zero posto

a uno.

jr z,LINEA0

– Il valore del flag di zero viene letto, e nel caso sia a 1 viene effettuato un salto relativo all'operazione contrassegnata dall'etichetta LINEA0.

ld b,a

– Il registro b viene caricato con il numero di riga. Piloterà il ciclo di somma ripetuta, che realizza la moltiplicazione 32*R (vedi sopra)

CICLO add hl,32

– Si tratta della somma, ripetuta per tutto il ciclo. Vediamo qui per la prima volta un'altra delle cose particolari che solo hl può fare: avere lo stesso ruolo di a come accumulatore, con numeri a 16 bit. 32, di per sé, potrebbe anche essere contenuto in un unico registro, ma non si può dire lo stesso per il numero già presente in hl (l'altro addendo), né per il risultato finale.

djnz CICLO

– È la chiusura del ciclo. La sua meccanica di funzionamento spiega perché non sia possibile entrare nel ciclo con b=0. In questo caso, infatti, dopo una prima somma (già di troppo) ne seguirebbero ben 255 altre. Perché? Semplice: se b=0, b-1=255 al primo passaggio per djnz.

LINEA0 ld a,(24103)

– Viene caricato il numero di colonna. Nel caso di R=0 l'esecuzione salta qui.

ld e,a
ld d,0

– Il valore di a viene trasferito in de. Questa operazione è necessaria per poter sommare ad hl il contenuto di a. Non è infatti possibile sommare ad hl un registro singolo.

add hl,de

– Si tratta di un'altra forma dell'addizione a 16 bit. Abbiamo già visto

add hl,NN

dove NN è un numero a 16 bit (sempre, anche se è piccolo; 32 va infatti espresso con due byte: 0 32). Questa è invece la forma

add hl, <coppia >

Il risultato va a finire in hl.

ld a,(24101)

– Caricamento del codice del carattere.

ld(hl),a

– Caricamento all'indirizzo contenuto in hl (è l'indirizzo finale) del codice.

Visto che l'ultimo dato è il codice di un carattere, può essere interessante modificare il programmino BASIC per renderne più semplice l'introduzione:

```
10 FOR i=24101 TO 24102
20 INPUT a
30 POKE i,a
40 NEXT i
50 INPUT a$
60 POKE i,CODE a$
```

sfruttando il fatto che al termine di un ciclo FOR...NEXT la variabile di controllo (i nel nostro caso) risulta maggiore di uno STEP rispetto al limite superiore del ciclo. Quindi

24102+1=24103

(lo STEP, se non è specificato, si intende 1).

Con questo nuovo programma è possibile dare direttamente il carattere che si vuole stampare, senza doverne conoscere il codice.

Il sistema di trasferimento dati appena visto funziona sia su ZX81 che su Spectrum. Quest'ultimo, però, dispone anche di un altro sistema, più sofisticato ed elegante.

Tutti gli "spectrumisti" sanno che la loro macchina dà la possibilità di creare funzioni "su misura", per mezzo dell'istruzione

DEF FN a(x,y,z)=...

cui segue la definizione della funzione desiderata. Le variabili fra parentesi sono, come è noto, variabili fittizie. Al momento dell'uso verranno cioè sostituite dai valori delle costanti e/o variabili presenti nell'istruzione

...FN a(1,2,h)

Nell'esempio, ad x sarà sostituito il valore 1, ad y il 2 e a z il valore della variabile h. In questo caso "sostituito" non è un modo di dire; infatti i valori

richiesti vengono materialmente inseriti in appositi spazi all'interno dell'istruzione DEF FN.

Inoltre, per permettere al programma un veloce reperimento della funzione richiesta, l'indirizzo di inizio della stessa istruzione viene indicato dalla variabile di sistema

DEFADD

La disposizione dei valori all'interno di un'istruzione DEF FN, al momento del suo richiamo da parte di una FN, è la seguente:

```
DEF FN a(x0E12345, y0E12345, z0E12345)
```

supponendo che le variabile siano tre, come nell'esempio. Abbiamo cioè un byte occupato dal carattere della variabile (x, y o z), cui segue un byte contenente il valore OE (14 decimale). Quindi ci sono cinque byte contenenti il valore attribuito alla variabile dalla FN, nella forma in cui lo Spectrum conserva i numeri. Senza inoltrarci nel problema della rappresentazione dei numeri decimali, a noi basta sapere che un numero intero e compreso fra 0 e 255 (è il genere di numeri che ci interessa in questo

momento) è contenuto tutto nel terzo byte.

Sapendo che il valore contenuto in DEFADD è l'indirizzo del byte che contiene il carattere della prima variabile, per recuperare il primo numero basterà cercarlo all'indirizzo

```
I1=(DEFADD)+4
```

il secondo, poi, sarà all'indirizzo

```
I2=I1+8
```

(occorre saltare due byte del numero, un byte per la virgola, uno per il carattere della seconda variabile, uno per il byte a 0E e due per i primi byte del nuovo numero), ed il terzo

```
I3=I2+8
```

Se ci fossero altre variabili, la cosa potrebbe andare avanti.

Quindi, se scriviamo un'istruzione tipo

```
DEF FN a(x,y,z)=USR <indirizzo >
```

che richiameremo con

```
...FN a(1,2,h)
```

la routine in l/m ad <indirizzo > avrà

a disposizione i dati attribuiti alla DEF FN dalla FN. Questo purché sappia come recuperarli, ma abbiamo già visto come si fa. Tradotto in Assembly può essere

```
ld hl,(DEFADD)
```

```
ld de,4  
add hl,de  
ld a,(hl)  
add hl,de  
add hl,de  
ld b,(hl)  
add hl,de  
add hl,de  
ld c,(hl)
```

Avremo in a il primo valore, in b il secondo, in c il terzo.

Questo sistema consente di passare alla routine in l/m anche numeri contenuti in variabili. Infatti è possibile scrivere

```
..FN a(a,b,c)
```

La prossima volta ne vedremo un esempio pratico.

Si tratterà di una routine che, dati il valore di riga e colonna ed il valore attribuiti, modifica gli attributi del carattere corrispondente.

A presto!

MI.PE.CO.

VENDITA PER
CORRISPONDENZA

SPECTRUM 48K PLUS 399.000

TUTTO COMPRESO
6 MESI
DI GARANZIA

La confezione comprende: alimentatore, manuale in inglese, **manuale in italiano**, cassetta dimostrativa, cavetti di collegamento e **istruzioni in italiano per la tastiera**
In regalo 5 programmi in italiano tra i quali il famoso "supercopiatore" di Massimo Rossi

QL ULTIMA VERSIONE CON NUOVI

PROGRAMMI 1.099.000
alimentatore manuale in inglese - 8 cartucce con 4 programmi

STAMPANTE ALPHACOM 32 199000

Per Spectrum e ZX 81 - istruzioni italiano - un rullo di carta in dotazione

STAMPANTE EPSON RX 80 699.000

con interfaccia centronics manuale in inglese

STAMPANTE MANNESMAN TALLY

MD 80+ 599.000

100 cps foglio singolo e modulo continuo - interfaccia centronics

CONVERTITORE DA RS232 A CENTRONICS

PER QL CON CAVO ADATTO L. 99.000

8 cartucce per microdrive 49000

AVVERTENZE

- tutti i prezzi sono comprensivi di IVA e spese postali
- pagamento contrassegno al ricevimento del pacco
- segreteria telefonica in funzione fuori orario, chiedete listini o altre informazioni; Vi risponderemo
- Sconti quantità
- Parti di ricambio per Spectrum
- Per ordini inferiori a L. 50.000 aggiungere L. 5.000 per spese postali.

TASTIERA DELLO SPECTRUM PLUS ... 79.000

kit per trasformare lo spectrum normale in Plus

ESTENSIONE PER SPECTRUM... 299000

La confezione comprende:

Microdrive

interfaccia uno



4 CARTUCCE CON 5 PROGRAMMI

introductory-master file-trasword TWO ant attack-gamesdesigner

- manuali in inglese
- cavetti di collegamento
- istruzioni in italiano

GARANZIA

La MI.PE.CO. si impegna a sostituire tutto il materiale trovato malfunzionante entro 48 ore dal ricevimento

INFORMAZIONI E ORDINI: MI.PE.CO. - C.P. 3016 - 00121 ROMA (OSTIA) - Tel. 06/5611251

Radar

di **Bill Lythgoe**
trad. e adatt.
di **Angela Spinazzola**

Gli alieni appaiono a frotte sul vostro radar. Solo dei riflessi eccezionali potranno salvare la Terra!

Sembra incredibile, ma dopo le stragi di alieni già compiute sugli schermi di tutto il mondo con migliaia di videogiochi, dai più profondi recessi del cosmo ancora giungono manipoli di extraterrestri alla conquista del nostro pianeta (ma lo sanno che è inquinato e litigioso?). Eccovi dunque di fronte allo schermo radar, in vigile attesa, per individuare e distruggere gli invasori. Siate scattanti, perché il tempo a vostra disposizione per attivare i pulsanti distruttivi diminuisce inesorabilmente. Il gioco terminerà quando 10 invasori saranno sfuggiti ai vostri mezzi di difesa (e speriamo per nostra buona sorte che essi incappino in qualche altro videogioco). La posizione dei nemici sul vostro schermo radar verrà individuata da due coordinate, contraddistinte da una lettera e da un numero. Ricordatevi di premere prima il tasto corrispondente alla lettera, poi quello relativo al numero. Ogni dieci alieni distrutti il bordo cambia colore e il tempo a vostra disposizione per rispondere diminuisce.

Descrizione del programma

Linee	Commento
90-180	preparazione dello schermo
210-250	scelta delle coordinate e stampa dell'alieno in posizione A
240	inizio del ciclo per la lettura della prima coordinata
255	invio alla subroutine per la lettura della tastiera
260	stampa dell'alieno in posizione B. Le posizioni A e B, alternandosi, danno l'illusione del movimento
280	ripetizione del ciclo aperto alla linea 240. Dopo 11 cicli, se l'alieno non è stato distrutto, viene conteggiato tra quelli mancati (t=t+1)
284	se gli alieni mancati sono 10, la partita ha termine
300	inizio del ciclo per la lettura della seconda coordinata
310-330	stampa dell'alieno nelle due posizioni che, alternandosi, danno l'illusione del movimento
320	invio alla subroutine per la lettura della tastiera
352	si conteggiano gli alieni mancati (t=t+1) e...
356	...se sono 10, il gioco ha fine
370	il giocatore ha saputo premere i tasti con le coordinate dell'invasore nel tempo prefissato. Il nemico è annientato...
375	...e viene cancellato
376-377	aggiornamento del punteggio
390	ogni dieci alieni distrutti...
405	...il tempo a disposizione per l'intervento viene diminuito. Cambia il colore del bordo (variabile q). Chi vuole variare i tempi di risposta deve intervenire su questa linea, cambiando il valore di "1". Se "1" viene moltiplicato per un coefficiente superiore a quello che compare nel listato, il tempo aumenta. Si può ottenere lo stesso risultato intervenendo sul valore dei BEEP che si trovano tra la linea 250 e la linea 330
410	ciclo di chiusura. È possibile riprendere per un'altra partita
1002-1021	lettura della tastiera per la prima coordinata, individuata dalle lettere "a" ... "j"
1101-1120	lettura della tastiera per la seconda coordinata, individuata dai numeri da zero a 9
1200-1290	caricamento dei caratteri grafici
9000-9990	linea di lancio. Comandi di registrazione e di verifica

Radar versione per ZX Spectrum 16K/48K

```
10 GO SUB 1200
20 LET q=2
```

```
30 BORDER 1: PAPER 1: INK 7: CLS
40 PRINT INK 6; AT 1,0;"AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA"; INK 7;
AT 3,11;"R A D A R"
41 PRINT INK 6; AT 5,0;"AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA"
```

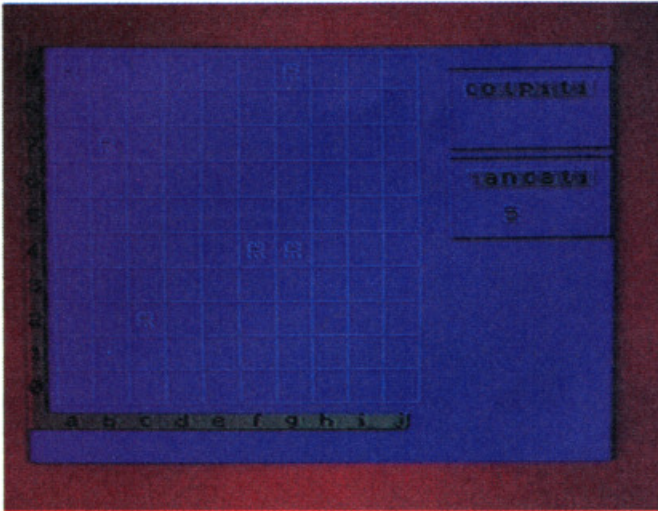
```

50 BEEP .2,7: BEEP 1.4,5
51 PRINT "'Hai avuto l'incarico di
  esplora-re il cielo con il tuo R
  ADAR.'"'"C'e' infatti la possibi
  lita' chedallo spazio giungano d
  egli es- seri misteriosi.'"'"Qua
  ndo li vedrai comparire do- vra
  i distruggerli premendo i ta-sti
  delle loro coordinate."
55 PRINT "'ATTENZIONE: premi prima
  il tasto          con la lette
  ra,              poi quello c
  on il            numero."
60 PRINT #0; PAPER 6; INK 0;"ABABAB
  AB premi un tasto ABABABAB"
75 BEEP .2,12: BEEP 1.4,10
80 PAUSE 0: INPUT ""
85 CLS
90 LET l=1: LET s=0: LET t=0
95 FOR a=12 TO 172 STEP 16
100 PLOT 12,a: DRAW 160,0
110 NEXT a
120 FOR b=12 TO 172 STEP 16
130 PLOT b,12: DRAW 0,160
140 NEXT b
150 PRINT PAPER 6; INK 0;" "'9"'"
  "'8"'" "'7"'" "'6"'" "'5"'"
  "'4"'" "'3"'" "'2"'" "'1"'"
  "'0"'" "' a b c d e f g h i j
  "
151 PLOT INK 6;185,165: DRAW INK 6
  ;68,0: DRAW INK 6;0,-36: DRAW
  INK 6;-68,0: DRAW INK 6;0,36
153 PRINT PAPER 7; INK 0;AT 2,24;"c
  olpiti"
154 PLOT INK 6;185,125: DRAW INK 6
  ;68,0: DRAW INK 6;0,-36: DRAW
  INK 6;-68,0: DRAW INK 6;0,36
155 PRINT AT 7,24; PAPER 7; INK 0;"m
  ancati"
160 IF q=8 THEN LET q=2
180 BORDER q
210 LET c=(((INT (RND*10))+0.5)*2)
220 LET d=(((INT (RND*10))+1)*2)
230 LET f=RND*12
240 FOR x=0 TO 10
250 PRINT AT c,d;"A": BEEP .15*1,f
255 GO SUB 1000+d
260 PRINT AT c,d;"B": BEEP .2*1,f-2
280 NEXT x
282 LET t=t+1
283 PRINT PAPER 7; INK 0;AT 9,26;t
284 IF t=10 THEN GO TO 410
285 GO TO 200
300 FOR y=0 TO 10
310 PRINT AT c,d;"A": BEEP .2*1,f
320 GO SUB 1100+c
330 PRINT AT c,d;"B": BEEP .2*1,f-2
350 NEXT y
352 LET t=t+1
354 PRINT PAPER 7; INK 0;AT 9,26;t
356 IF t=10 THEN GO TO 410
360 GO TO 200
370 PRINT AT c,d;"P": FOR n=12 TO -1
  2 STEP -1: BEEP .01,n: NEXT n
375 PRINT AT c,d;" "
376 LET s=s+1
377 PRINT PAPER 7; INK 0;AT 4,26;s
390 IF (s/10)-INT (s/10)=0 THEN
  GO TO 405
400 GO TO 200
405 LET l=0.5*l: LET q=q+1: GO TO 16
  0
410 GO SUB 500: PRINT PAPER 6;
  INK 0;AT 16,24;"GIOCHI ";AT 17,
  24;"ANCORA?";AT 18,24;" (S/N) ":
  INPUT a$
420 IF a$="s" THEN CLS : GO TO 85
430 STOP
500 FOR y=1 TO 19 STEP 2
510 FOR x=2 TO 20 STEP 2
520 PRINT AT y,x; INK 2+RND*5;"A"
530 BEEP .005,x
540 NEXT x
550 NEXT y
560 RETURN
1002 IF INKEY$="a" THEN GO TO 290
1003 RETURN
1004 IF INKEY$="b" THEN GO TO 290
1005 RETURN
1006 IF INKEY$="c" THEN GO TO 290
1007 RETURN
1008 IF INKEY$="d" THEN GO TO 290
1009 RETURN
1010 IF INKEY$="e" THEN GO TO 290
1011 RETURN
1012 IF INKEY$="f" THEN GO TO 290
1013 RETURN
1014 IF INKEY$="g" THEN GO TO 290
1015 RETURN
1016 IF INKEY$="h" THEN GO TO 290
1017 RETURN
1018 IF INKEY$="i" THEN GO TO 290
1019 RETURN
1020 IF INKEY$="j" THEN GO TO 290
1021 RETURN
1101 IF INKEY$="9" THEN GO TO 370
1102 RETURN
1103 IF INKEY$="8" THEN GO TO 370
1104 RETURN
1105 IF INKEY$="7" THEN GO TO 370
1106 RETURN
1107 IF INKEY$="6" THEN GO TO 370
1108 RETURN
1109 IF INKEY$="5" THEN GO TO 370
1110 RETURN
1111 IF INKEY$="4" THEN GO TO 370
1112 RETURN
1113 IF INKEY$="3" THEN GO TO 370
1114 RETURN

```

LOAD

Radar



```

1115 IF INKEY$="2" THEN GO TO 370
1116 RETURN
1117 IF INKEY$="1" THEN GO TO 370
1118 RETURN
1119 IF INKEY$="0" THEN GO TO 370
1120 RETURN
1200 FOR n=1 TO 3: READ p$
1210 FOR f=0 TO 7
1220 READ a: POKE USR p$+f,a
1230 NEXT f
1240 NEXT n
1250 DATA "a",BIN 01111110,BIN 1101
1011,BIN 11111111,BIN 11000011
,BIN 01111111,BIN 01001001,
BIN 11011011,BIN 10010010
1260 DATA "b",BIN 01111110,BIN 1101
1011,BIN 11111111,BIN 11000011
,BIN 11111110,BIN 10010010,
BIN 11011011,BIN 01001001
1270 DATA "p",BIN 10010001,BIN 0100
0010,BIN 00100100,BIN 00000001
,BIN 10000000,BIN 00100100,
BIN 01000010,BIN 10001001
1280 RETURN
9000 CLS : PRINT FLASH 1;AT 10,0;"
{32G3} FERMA IL NASTRO
{32SG3}"
9010 PRINT AT 21,0;"Premi un tasto pe
r incominciare.": PAUSE 0: RUN
9990 CLS : SAVE "radar" LINE 9000
9992 VERIFY ""
9999 STOP

```

Per ricevere le riviste arretrate
di SUPERSINC con cassetta
compilate e spedite il coupon qui sotto!

BUONO D'ORDINE PER RIVISTE ARRETRATE

Anche se l'ordine riguardasse la sola cassetta questa
verrà comunque inviata insieme alla rivista al prezzo qui
indicato.

SUPERSINC

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA
da compilare e spedire in busta chiusa a
J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

Vogliate inviarmi i numeri
di SUPERSINC con cassetta
Al prezzo di L. 15.000

Nome Anno

Cognome Prov.

Via C.A.P.

Città Se richiesta fattura:
Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul cc. n° 19445204 intestato a J. soft - Milano

Pagherò in contrassegno al postino al ricevimento

Le modifiche definitive al modello 740

Sebbene siano ormai inutili ai fini di un utilizzo pratico del programma (ma qualcuno potrebbe pensare, magari, di riadattarlo in vista della dichiarazione dei redditi 1986...), pubblichiamo ugualmente alcune modifiche da apportare al programma per la compilazione del modello 740, compreso su SUPERSINC di Maggio 1985:

- 1) la linea 150 del sottoprogramma "740/84 1" va - se ancora non lo avete fatto - modificata secondo quanto suggerito nell'errata corrige pubblicato sul numero scorso della rivista:
150 LET o\$="":IF t<>VAL "2" THEN GO TO VAL "240"
- 2) la linea 5215 del sottoprogramma "740/84 2" va modificata come segue, e NON come è stato scritto nel numero scorso - alcuni "regali" della fotocomposizione hanno introdotto degli errori nell'...errata corrige:
5215 IF con=o THEN LET es2=o: LET co2=o: LET fc2=o: LET ac2=o: LET ud2=o: GO TO VAL "5245"
- 3) sempre nel "740/84 2" va aggiunta la linea 5421 che riportiamo di seguito. Questa linea sostituisce completamente le correzioni che venivano richieste per le linee 5350 e 5360, le quali vanno lasciate esattamente com'erano. Per fare chiarezza, riportiamo anche l'aspetto definitivo delle linee in oggetto:
5421 IF cz>VAL "1474.201" THEN LET uddl=o
5350 IF c1+cc1=o OR c1+cc1>VAL "17690" THEN LET uddl1=o: GO TO VAL "5485"
5360 IF c2+cc2=o OR c2+cc2>VAL "17690" THEN LET uddl2=o: GO TO VAL "5500"
- 4) nelle linee 1865 e 1870 (sempre di "740/84 2") tutte le z vanno sostituite con x.
- 5) Queste non sono correzioni di possibili bug del programma, ma solo migliorie (vale a dire che è possibile saltarle senza problemi).

```
4950 LET on2=int2+cure2+ass2+alt  
ri2: LET oneri2=ilor2+on2: IF rc  
2<=oneri2 THEN LET ri2=o: LET il  
2=o: GO TO VAL "4975"
```

←4975 anziché 5000

```
4975 LET res1=o: IF ri1>VAL "100  
00" THEN GO TO VAL "4985"
```

```
4976 INPUT "il DICHIARANTE ha re  
dditi esentio con ritenute alla  
fonte? (s/n)":;s$: IF s$="n" THE  
N GO TO VAL "4985"
```

```
4980 IF s$<>"s" THEN GO TO VAL "  
4976"
```

```
4981 INPUT "totale redditi esent  
i o con rit.alla fonte DICHIARAN  
TE: ";res1: IF res1<=VAL "2000"  
THEN LET res1=o
```

←Aggiungere linee da 4975 a 4991

```
4985 LET res2=o: IF (rc2=o AND c  
m=o) OR ri2>VAL "10000" THEN GO  
TO VAL "5000"
```

```
4986 INPUT "il CONIUGE ha reddi  
i esenti o con ritenute alla fo  
nte? (s/n) ";s$: IF s$="n" THE  
N GO TO VAL "5000"
```

```
4990 IF s$<>"s" THEN GO TO VAL "  
4986"
```

```
4991 INPUT "totale redditi esent  
i o con rit.alla fonte CONIUGE:  
";res2: IF res2<=VAL "2000" THEN  
LET res2=o
```

```
5100 LET es1=VAL "96": IF ri1+re  
s1>VAL "10000" THEN LET es1=VAL  
"36"
```

Aggiungere:

←+ res 1

```
5110 IF (rc2=o AND cm=o) OR rc2+  
res2>VAL "2750" THEN LET co1=o:  
GO TO VAL "5120"
```

←+ res 2

```
5111 IF con=l AND cm=o AND rc2+r  
es2<=VAL "2750" THEN INPUT "il c  
oniuge in effetti e'a caricoper  
mesi: ";cm
```

←+ res 3

```
5185 IF t=VAL "2" OR t=VAL "4" O  
R t=VAL "5" THEN GO TO VAL "5200  
"
```

←VAL "2" or

```
5220 LET es2=VAL "96": IF ri2+re  
s2>VAL "10000" THEN LET es2=VAL  
"36"
```

←+ res 2

```
5225 IF co1>o OR rc1+res1>VAL "2  
750" THEN LET co2=o: GO TO VAL "  
5235"
```

←+ res 1

Ultima cosa: il programma è costituito da 4 blocchi: di seguito vi diamo le istruzioni di SAVE per ognuno di essi:

"740 a" LINE 1

"740 c" CODE 63833,1688

"740/84 1" LINE 1

"740/84 2" LINE 990

Ricordatevi sempre di:

- a) dare un NEW prima di eseguire le modifiche, caricando il necessario sottoprogramma con MERGE"
- b) dare SEMPRE un CLEAR prima di registrare il sottoprogramma modificato.



```

30 PRINT PAPER 7;AT 13,4;" Usa i t
asti ~Q~ e ~P~ "
35 PRINT "" Se finisci lo spray sar
ai co- stretto a subire l'assal
to delle loro larve chiuse in un
disco volante."
40 PRINT AT 21,9;"PREMI UN TASTO"
60 IF INKEY$="" THEN GO TO 60
80 BORDER 2: PAPER 5: INK 0: CLS
90 LET b=0: LET s=0: LET c=50
100 LET e=29
120 PRINT INK 1;AT 21,0;"{2SG8}
{8SG3}{2SG8}{18SG3}";c
130 PRINT INK 1;AT 20,0;"IE";AT 20,
10;"IE"
140 LET y=INT (RND*17)+2
145 PRINT AT 21,30;" ";AT 21,30;c
150 PAUSE 30: BEEP .1,20
160 LET n=y: LET y=y+(INT (RND*3)-1)
-(19=y)+(1=y)
170 PRINT AT n,e;" ": PRINT INK 2
;AT y,e;"GH"
180 IF c<=0 THEN PRINT OVER 1;AT y
,e;"GH": GO TO 600
220 IF INKEY$="q" THEN PRINT AT 20,
0; INK 1;"J": PLOT 17,17: DRAW 1
50,150: BEEP .05,36: PLOT
OVER 1;17,17: DRAW OVER 1;150,
150: PRINT AT 20,0; INK 1;"I":
LET c=c-1: IF 21-y=e OR 21-y=e+
1 THEN GO TO 400
230 IF INKEY$="p" THEN PRINT AT 20,
10; INK 1;"J": PLOT 97,17:
DRAW 150,150: BEEP .05,36:
PLOT OVER 1;97,17: DRAW
OVER 1;150,150: PRINT AT 20,10;
INK 1;"I": LET c=c-1: IF 31-y=e
OR 31-y=e+1 THEN GO TO 400
240 BEEP .015,-12
250 IF e=0 THEN PRINT AT y,e;" ":
GO TO 410

```

```

260 LET e=e-INT (RND*2)
280 GO TO 160
400 LET s=s+1: PRINT INK 0;AT y,e-1
;"CDCD": BEEP .3,-25: PAUSE 20:
PRINT AT y,e-1;" "
410 LET b=b+1: LET e=29
420 IF b>19 THEN CLS : GO TO 440
430 GO TO 140
440 PRINT PAPER 7;AT 6,7;"Fine dell
a partita"
450 PRINT PAPER 6;AT 8,6;"I TUOI PU
NTI SONO ";s;AT 10,4;"realizzati
con ";50-c;" getti"
455 PRINT PAPER 7; INK 2;AT 20,0;"~
G~ per giocare - ~F~ per finire"

460 IF s=18 OR s=19 THEN PRINT AT 1
5,0;"Non e' male il punteggio pu
r es-sendo tu un principiante."
470 IF s=20 THEN PRINT AT 15,0;" C
OMPLIMENTI! SEI (quasi) DA MED
AGLIA OLIMPICA!"
480 IF s<6 THEN PRINT AT 15,0;"Spia
cente, ma sei una schiappa."
490 FOR f=-10 TO 10: BEEP .1,f:
BORDER RND*7: NEXT f: BEEP 2,-1
7: BORDER 7
500 IF INKEY$="" THEN GO TO 500
510 IF INKEY$="g" THEN GO TO 80
520 PRINT FLASH 1; PAPER 6;AT 21,0;
" A R R I V E D E R C I
": STOP
600 LET e=29: PRINT AT 0,0; FLASH 1;
" POVERO TE! LO SPRAY E' FINITO
"
610 LET e=e-1
620 PRINT INK 2;AT 18,e;"AB ":
FOR i=1 TO 6: BEEP .015,-12:
PAUSE 2: NEXT i
630 IF e=22 THEN PLOT 184,25:
DRAW -108,-20: PAUSE 3: PLOT
OVER 1;184,25: DRAW OVER 1;-10
8,-20
640 IF e=21 THEN PRINT AT 20,9;"FDC
F";AT 19,9;"FCDF";AT 21,10;"FF"
643 BEEP .015,-12
645 IF e=21 THEN FOR i=1 TO 20:
BEEP .01,i/4: BORDER INT (i/3):
NEXT i
650 IF e=12 THEN PLOT 104,25:
DRAW -104,-20: PAUSE 3: PLOT
OVER 1;104,25: DRAW OVER 1;-10
4,-20
660 IF e=11 THEN PRINT AT 20,0;"DCF
";AT 19,0;"DF";AT 21,0;"CF"
665 IF e=11 THEN FOR i=1 TO 20:
BEEP .01,i/4: BORDER INT (i/3):
NEXT i
670 IF e=6 THEN CLS : PRINT AT 2,0;
"Hai perso e le larve sono state
inesorabili": GO TO 440
680 GO TO 610
1000 FOR f=0 TO 79
1010 READ cg: POKE USR "a"+f,cg:
NEXT f

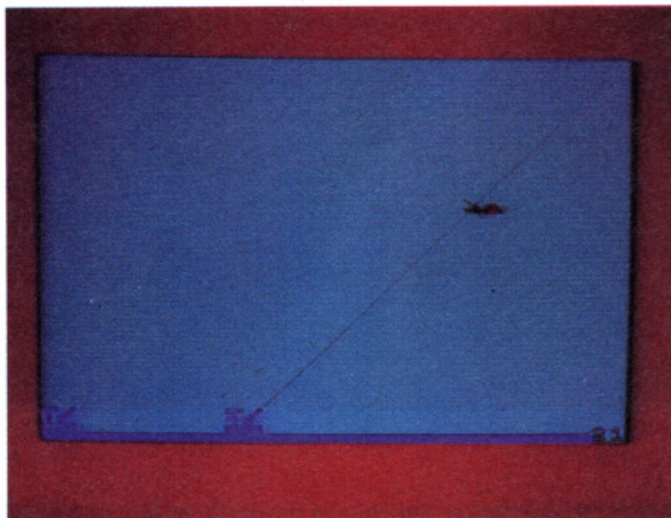
```

LOAD

Zanzare

```

1090 DATA 0,0,15,63,127,255,255,3,0,0
      ,240,252,254,255,255,192
2000 DATA 130,16,0,66,0,36,0,72,65,8,
      0,66,0,36,0,18
2010 DATA 3,7,14,60,248,248,240,240,0
      ,32,0,4,64,0,16,0
2020 DATA 128,32,12,30,255,6,0,1,0,0,
      60,126,255,124,0,64
2030 RETURN
2040 DATA 255,255,255,24,24,24,24,24,
      0,0,0,0,255,255,255,24
9000 CLS : PRINT FLASH 1;AT 10,0;"
      FERMA IL NASTRO "
9010 PRINT AT 21,0;"Premi un tasto pe
r incominciare.": PAUSE 0: RUN
9990 CLS : SAVE "zanzare" LINE 9000
9995 PRINT "SAVE completata. Prepara
il nastro per VERIFY.":
      VERIFY ""
9999 PRINT ""Programma O.K.! Ciao.":
      STOP
  
```



ERRATA CORRIGE

Nel programma "contapersone" pubblicato nella rubrica REM:HW di SUPERSINC numero 3 (marzo 1985) la linea 1015 è superflua, avendo solamente funzioni di controllo. In ogni caso il programma funziona correttamente, presentando lievi disturbi nella sola fase di visualizzazione. Ci scusiamo con tutti i lettori per l'inconveniente.

La redazione

Novità firmate Jackson.

CON CASSETTA



Michel Benéfouf
GIOCHI, TRUCCHI E CONTI

Questo libro è destinato soprattutto ai programmatori principianti: con i suoi 29 programmi presentati con una ricca documentazione e pronti per essere eseguiti. Gli argomenti sono tanti e rispondono alle esigenze più disparate. Il BASIC è quello dei calcolatori PET/CBM di cui vengono sfruttate a pieno le molte funzioni.

Cod. 553D Pag. 164 Lire. 24.000

Rita Bonelli
Luciano Pazzuconi - Fabio Racchi
Giovanni Valerio
**COMMODORE 64
LA GRAFICA E IL SUONO**

Ogni argomento viene spiegato e accompagnato da numerosi esempi commentati. Nel libro sono listati moltissimi programmi; essi sono registrati sulla cassetta allegata. Inoltre la cassetta contiene anche altri programmi, ai quali si fa riferimento nel testo, ma senza listarli.
Cod. 409B Pag. 270 Lire. 34.000



**GRUPPO
EDITORIALE
JACKSON**

ritagliare (o fotocopiare) e spedire in busta chiusa a:

GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDEOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

N° _____ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome _____

Cognome _____

Via _____

Cap _____ Città _____ Prov. _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

ORDINE
MINIMO
L. 50.000

Partita I.V.A. _____

Raccolta nel frutteto

di **Angus Lavery**
trad. e adatt.
di **Angela Spinazzola**

Un simpatico gioco per sviluppare i riflessi dei più (e meno) piccoli

Uno dei motivi per cui la frutta è molto cara viene imputato al costo della mano d'opera necessaria per la raccolta. È vero che la meccanizzazione ha fatto grandi passi avanti in proposito, ma l'opera dell'uomo è ancora estremamente necessaria. D'altra parte aggirarsi tra gli alberi col cesto sottobraccio alla ricerca di profumati frutti di stagione credo sia attività gradita a tutti. Andiamo dunque con il nostro fedele Spectrum nel frutteto, un frutteto tutto particolare nel quale i frutti cadono da soli. Bisogna essere molto svelti a spostare il cestino nel punto giusto, perché ogni frutto raccolto ha un valore e noi dobbiamo cercare di guadagnare più soldi possibile nell'arco della partita. Non bisogna certo dormire sotto le fronde! Le cadute sono rapide e vengono stabilite dalla funzione RND, pertanto sono imprevedibili. Il cesto non è gigantesco ed è facile farlo scivolare troppo avanti od arrestarlo troppo presto.

Un guadagno complessivo superiore alle duemila lire per partita è da ritenersi molto buono; in casa mia l'impresa è riuscita solo a mia figlia. Non tutti i frutti hanno lo stesso valore, ma non è concesso trascurarne alcuno, nemmeno quelli da cento lire.



Descrizione del programma

Linee

10-63
70-100
110-240
250
270
290-350
370-380
382-383

Commento

presentazione del gioco
valore dei frutti e tasti da usare
stampa del frutteto
inizializzazione delle variabili di gioco
con la funzione RND viene scelto il frutto da far cadere
caduta del frutto
stampa del cestello per la raccolta e movimento dello stesso
controlla se il frutto è caduto nel cesto. In caso positivo incrementa il punteggio, quindi ritorna all'inizio del ciclo di gioco
stampa la somma guadagnata...
... e se la stessa costituisce il nuovo record del giocatore, ne dà avviso ed aggiorna il record
possibilità di rifare la partita o, se volete, la raccolta
carica i caratteri grafici necessari al gioco
linea di lancio
comandi di SAVE

410-450
465

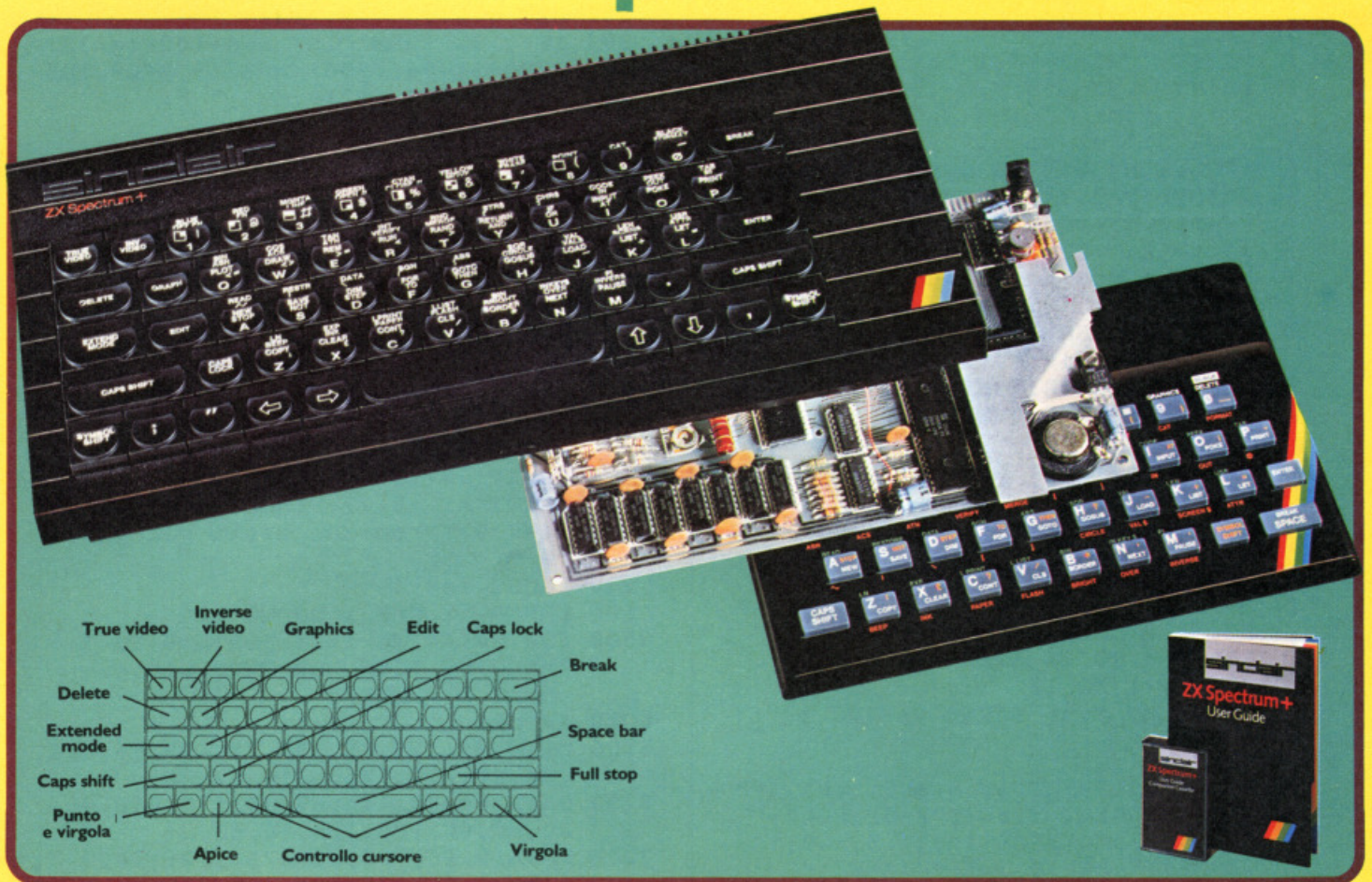
470-500
8000
9000
9990

Raccolta nel frutteto versione per ZX Spectrum 16K/48K

```
10 LET hw=0
20 BORDER 1: PAPER 5: INK 0: CLS
30 GO SUB 8000
```

```
40 PRINT PAPER 6;AT 1,0;"#####
# FRUTTETO #####"
50 PRINT "' BRIGHT 1;"Istruzioni"
55 PRINT '"Sei stato assunto in un
fruttetoper raccogliere i frutt
i che ca-dono dagli alberi. Non
te ne de-vi far sfuggire nemmeno
```

Trasforma il tuo Spectrum in ZX Spectrum +



Ecco una novità stimolante per i possessori di Spectrum :
 Il KIT ORIGINALE SINCLAIR, che promuove lo Spectrum al grado superiore.
 Non si richiede vasta esperienza . Basta saper saldare pochi fili.

CARATTERISTICHE:

- Tastiera professionale SINCLAIR con 17 tasti extra.
- Si usa come una normale macchina da scrivere.
- Compatibile con tutto il software e le periferiche Spectrum.
- Completo di una guida di 80 pagine più una cassetta dimostrativa.

**a casa
vostra subito !!**

Descrizione	Q.tà	Prezzo unitario	Prezzo Totale
Kit 48K/Plus		L. 109.000	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA
 Partita I.V.A.

PAGAMENTO:

- A) Anticipato, mediante assegno bancario per l'importo totale dell'ordinazione.
- B) Contro assegno, in questo caso, è indispensabile versare un acconto di almeno il 50% dell'importo totale mediante assegno bancario. Il saldo sarà regolato contro assegno.
- AGGIUNGERE: L. 5.000 per contributo fisso.
 I prezzi sono comprensivi di I.V.A.

DIVIS.

EXELCO

Via G. Verdi, 23/25
 20095 - CUSANO MILANINO - Milano

```

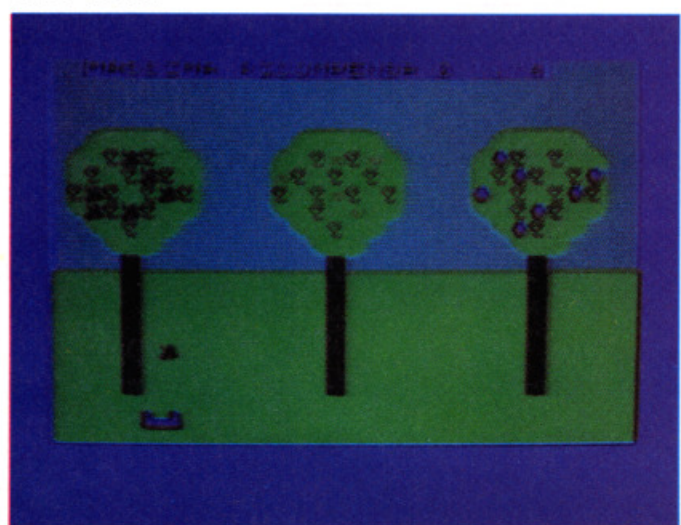
    uno se vuoi guadagnarti una bu
    ona paga."
62 FOR b=1 TO 3: FOR a=-40 TO 40:
    BEEP .01,a: BEEP .01,10-a:
    NEXT a: NEXT b
63 CLS
70 PRINT "Valore dei frutti"
75 PRINT INK 6;"TAB 12;"E"; INK 0
    ;" =1 punto" INK 2;TAB 12;"C";
    INK 0;" =2 punti" INK 1;TAB 12
    ;"D"; INK 0;" =3 punti"
76 PRINT 'TAB 7; PAPER 7; BRIGHT 1;
    "Tasti di controllo" "" 5 - sin
    istra ** 8 - destra "
80 PRINT "'Vuoi guadagnare una buon
    a paga?"
90 PRINT FLASH 1'"Premi un tasto p
    er incominciare."
95 FOR a=0 TO 40: BEEP .01,a
97 IF INKEY$<>" THEN CLS :
    GO TO 110
100 BEEP .01,10-a: NEXT a: GO TO 95
110 FOR a=12 TO 21
120 PRINT AT a,0; PAPER 4;"
    "
130 NEXT a
140 PRINT AT 0,2; PAPER 1; INK 6;"MA
    SSIMA RICOMPENSA ";hw;" lire"
150 FOR a=4 TO 26 STEP 11
160 FOR b=11 TO 18
170 PRINT AT b,a; PAPER 0; INK 2;"A"
180 NEXT b
190 PAPER 8: PRINT AT 4,a-2; INK 4;"
    F{3SG8}H";AT 5,a-3;"F{5SG8}H";
    AT 8,a-3;"G{5SG8}I";AT 9,a-2;"G
    {3SG8}I";AT 10,a-1;"G{SG8}I"
200 PRINT PAPER 4; INK 0;AT 5,a-2;"
    B B ";AT 6,a-3;" B B B ";AT 7,a
    -3;"B B B B";AT 8,a-2;" B B ";
    AT 9,a-1;" B ": NEXT a
220 PRINT AT 5,4; INK 2;"C";AT 6,5;"
    C";AT 7,2;"C";AT 7,6;"C";AT 8,2;
    "C";AT 8,4;"C"
230 PRINT AT 5,15; INK 6;"E";AT 5,17
    ;"E";AT 6,12;"E";AT 7,15;"E";
    AT 8,16;"E"
240 PRINT AT 5,24; INK 1;"D";AT 6,25
    ;"D";AT 6,29;"D";AT 7,23;"D";
    AT 7,28;"D";AT 8,26;"D";AT 9,25;
    "D"
250 LET s=0: LET f=15
270 LET c=INT (RND*6)+1
280 BEEP .02,.1
290 FOR a=10 TO 21
300 IF a>20 THEN GO TO 420
310 IF c=1 OR c=6 THEN PRINT AT a,6
    ; INK 2;"C";AT a-1,6;" "
320 IF c=2 OR c=5 THEN PRINT AT a,1
    6; INK 6;"E";AT a-1,16;" "
330 IF c=3 THEN PRINT AT a,12;
    INK 6;"E";AT a-1,12;" "
340 IF c=4 THEN PRINT AT a,22;
    INK 1;"D";AT a-1,22;" "
350 IF a=10 OR a=11 THEN PRINT AT a
    -1,6+10*(a-10); INK 4;"I"

```

```

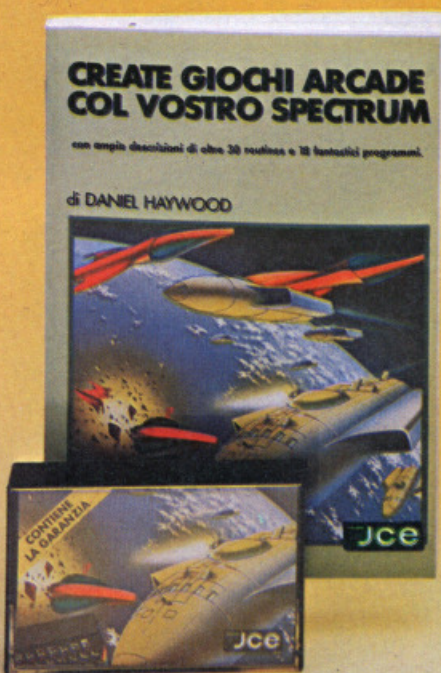
370 LET f=f+2*((INKEY$="8" AND f<25)
    -(INKEY$="5" AND f>0))
380 PRINT AT 20,f; INK 3;" {SG1}
    {SG2} "
382 LET g=ATTR (19,f+3)
383 IF g>32 THEN LET s=s+100*(g=38)
    +200*(g=34)+300*(g=33): BEEP .05
    ,10: PRINT AT 19,f+3;" ":
    GO TO 270
410 NEXT a
420 FOR a=1 TO 20
430 PRINT AT 21,0; PAPER 7; INK 1;"H
    ai guadagnato ";s;" lire"
440 BEEP .05,2: BEEP .01,-2
450 NEXT a
460 CLS
465 IF s>hw THEN LET hw=s: PRINT
    BRIGHT 1;AT 5,4;"E' AUMENTATA L
    A TUA PAGA": FOR x=1 TO 15:
    BEEP .01,x: BEEP .05,15-x:
    BORDER RND*7: BEEP .01,x:
    BEEP .01,30-x: NEXT x
470 PRINT AT 18,2; FLASH 0; PAPER 7;
    INK 0;"Premi un tasto per gioca
    re "
480 PAUSE 4e4
500 PAPER 5: CLS : GO TO 110
8000 FOR a=144 TO 152: FOR b=0 TO 7:
    READ c: POKE USR CHR$ a+b,c:
    NEXT b: NEXT a
8010 RETURN
8100 DATA 75,-75,75,-75,75,-75,75,-75
    ,42,84,170,68,40,16,42,20
8150 DATA 64,78,110,60,30,127,246,96,
    20,60,60,126,126,60,60,24
8200 DATA 0,24,126,255,255,126,24,0,3
    ,15,31,63,127,127,255,255
8250 DATA 255,255,127,127,63,31,15,3,
    192,240,248,252,254,254,255,255
8300 DATA 255,25,254,254,254,252,248,
    240,192
9000 CLS : PRINT FLASH 1;AT 9,0;"###
    #####
    FERMA IL NASTRO ###
    #####"
9010 PRINT AT 21,0;"premi un tasto pe
    r incominciare.": PAUSE 4e4:
    RUN
9990 SAVE "frutteto" LINE 9000
9995 VERIFY ""
9999 STOP

```



LIBRI

edizioni **Jce**



CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM

di DANIEL HAYWOOD

Il volume descrive dettagliatamente tutte le tecniche di stesura di giochi ARCADE, partendo dalla lettura della tastiera e toccando la definizione grafica, l'impiego del suono e l'uso degli operatori logici, per migliorare la qualità dei programmi. Altri argomenti esaminati in dettaglio sono l'animazione degli oggetti, lo scrolling dello schermo e l'impiego dei comandi PEEK e POKE per il loro uso più corretto. Il tutto accompagnato da 18 programmi la maggior parte dei quali o sono inediti, oppure riguardano versioni migliorate di games di grande successo come "INVADERS" e "BOMBER". I più interessanti sono stati registrati sulla cassetta allegata al volume al fine di farvi risparmiare ore di digitazione.

Pag. 116
Libro più cassetta

Cod. 9003 L. 25.000

BASIC & FORTRAN PER SPECTRUM

di S. J. WAINWRIGHT e A. GRANT

Questo libro può essere utilizzato per imparare sia il FORTRAN che il BASIC, od anche per apprendere entrambi i linguaggi contemporaneamente sul vostro SPECTRUM. La cronologia dei linguaggi dimostra quindi che il FORTRAN occupa una posizione di rilievo, soprattutto per ciò che concerne la programmazione di problemi scientifici e matematici. Mentre il BASIC è il più diffuso dei linguaggi di programmazione per microcomputer.

In questo libro e nella cassetta allegata è stato inserito un interprete FORTRAN per lo SPECTRUM nella massima configurazione (48k) che però con alcune piccole modifiche chiaramente indicate nel testo, potrà essere utilizzato con la versione inespansa ed anche sul SINCLAIR ZX 81.

Questo interprete vi aiuterà subito a comprendere i fondamenti della programmazione in FORTRAN, rendendo possibile la creazione di programmi che vi daranno una visione ben precisa delle potenziali caratteristiche di questo linguaggio.

Pag. 88
Libro più cassetta

Cod. 9007 L. 25.000

IMPARIAMO IL PASCAL SUL NOSTRO COMPUTER

di JEREMY RUSTON

Questo libro si rivolge a chi desidera conoscere il PASCAL ed apprenderne l'uso in modo semplice e lineare: è quindi adatto anche a chi è alle prime armi nel campo dell'informatica.

Nel libro sono riportati i listati di due programmi compilatori per tradurre le istruzioni PASCAL in BASIC: questo consente al lettore di provare direttamente programmi in PASCAL sul suo personal computer senza dover affrontare la spesa di un vero compilatore PASCAL.

Il primo compilatore è scritto in Basic MICROSOFT, quindi è adatto ai personal computer IBM PC; IBM compatibili, OLIVETTI M 10 - M 20 - M 21 - M 24 - HP 150. Il secondo è scritto in Basic SINCLAIR per lo ZX SPECTRUM ed è fornito su cassetta software allegata al libro.

Pag. 112
Libro più cassetta

Cod. 9800 L. 25.000

Descrizione	Cod.	Q.tà	Prezzo unitario	Prezzo Totale
CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM Libro più cassetta	9003		L. 25.000	
BASIC & FORTRAN PER SPECTRUM Libro più cassetta	9007		L. 25.000	
IMPARIAMO IL PASCAL SUL NOSTRO COMPUTER Libro più cassetta	9800		L. 25.000	

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA

Partita I.V.A.

PAGAMENTO:

Anticipato, mediante assegno bancario o vaglia postale per l'importo totale dell'ordinazione.

Contro assegno, al postino l'importo totale

AGGIUNGERE: L. 3.000 per contributo fisso spedizione. I prezzi sono comprensivi di I.V.A.

edizioni **Jce**

Via G. Verdi, 23/25
20095 - CUSANO MILANINO - MILANO

SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE



ALLA SCOPERTA DELL'APPLE IIc

di F. WAGNER DOBLER

Il libro, rivolto ad utilizzatori professionali e hobbistici, espone in dettaglio le disponibilità e le capacità dell'Apple IIc, incluso il suo hardware, le sue periferiche, le sue possibilità grafiche e l'uso di elaboratori di testo, tabelloni elettronici ed altre utilità software. Nel testo è compreso un esame delle specifiche hardware della macchina, un'analisi della documentazione ed una parte dedicata all'implementazione dei linguaggi di programmazione BASIC, LOGO, PASCAL, PILOT ed assembler 65C02. Il sistema operativo è trattato in dettaglio; cura particolare è stata dedicata al confronto fra l'Apple IIc e il suo predecessore Apple IIe. Il presente libro costituisce un'inesostituibile guida di riferimento per coloro che si propongono di acquistare l'Apple IIc e desiderano un obiettivo giudizio delle sue capacità, oppure per coloro che, già possedendolo, sono interessati ad ulteriori informazioni pratiche sulle sue possibili applicazioni.

Pag. 144

Cod. 9301 L. 16.000

IL 68000: PRINCIPI E PROGRAMMAZIONE

di LED J. SCANLON

Questo libro illustra le caratteristiche del microprocessore 68000 e fornisce le nozioni fondamentali per la sua corretta programmazione.

In particolare vengono illustrati:
 - il Cross MacroAssembler della Motorola
 - il set di istruzioni del 68000 con i suoi 14 modi di indirizzamento
 - le potenti istruzioni aritmetiche per moltiplicazione e divisione
 - programmi esemplificativi per la gestione delle liste e delle tavole numeriche.
 Vengono inoltre illustrate le funzioni dei 64 piedini del 68000 IC e i supporti periferici interfacciabili con esso. Conclude il volume una breve descrizione degli altri processori appartenenti alla famiglia del 68000, con particolare riguardo per il 68008, utilizzato nei personal computer dell'ultima generazione.

Pag. 256

Cod. 9850 L. 20.000

APPLE MACINTOSH: IL COMPUTER MAGICO

di E.S. CONNOLLY e P. LIEBERMAN

In un unico volume uno sguardo all'Apple Macintosh: lo strumento che farà salire vertiginosamente la vostra produttività nel lavoro.

Nel libro troverete:

- Come si può rendere più efficiente un calcolatore da tavolo
 - Come il MacPensiero incrementa la produttività
 - Che cosa sta dietro alle MacWindows
 - Come ottenere il massimo dal Mouse
 - L'uso di MacWriter, MacPaint e di tutti gli altri MacTools
 - Quale altro software è disponibile per il Macintosh
 - Come funziona il microprocessore 68000
 - Tutto sui drives per mini-floppy.
- Inoltre imparerete come comunicare con il Macintosh e come creare menù che parlino da soli.

Se ne possedete un esemplare o se intendete acquistarne uno, questa guida costituirà il vostro indispensabile e sapiente amico.

Pag. 192

Cod. 9350 L. 20.000

LA PRIMA VOLTA CON APPLE

di R. COOK e T. HARTNELL

Se non avete mai programmato un computer prima d'ora e vi piacerebbe imparare in poche ore, ecco il libro adatto a voi. Probabilmente già disporrete di gran parte, se non di tutto, del software che vi serve, ma vi sarete anche resi conto che è bene saperne un po' di più, magari per far colpo sugli amici, insegnare qualcosa ai vostri bambini o creare da soli qualche programma per lavoro o per gioco. Questo libro vi spiega come far tutto ciò in pochissimo tempo. All'inizio vedremo i pochi comandi che servono per lavorare con il disco di sistema, cioè per dare il via al computer.

Il tutto avverrà sotto il vostro controllo a partire dalle prima pagina del primo capitolo del libro. Passeremo poi in rassegna i termini più importanti per la programmazione e alla fine, avrete in mano una piccola raccolta di programmi che vi terranno occupati con Apple per qualche settimana.

Pag. 100

Cod. 9300 L. 16.000

Cedola di commissione libraria da inviare a: JCE - Via dei Lavoratori, 124 - 20092 Cinisello B. - MI

Descrizione	Cod.	Q.tà	Prezzo Unitario	Prezzo Totale
ALLA SCOPERTA DELL'APPLE IIc	9301		L. 16.000	
LA PRIMA VOLTA CON APPLE	9300		L. 16.000	
IL 68000: PRINCIPI E PROGRAMMAZIONE	9850		L. 20.000	
APPLE MACINTOSH: IL COMPUTER MAGICO	9350		L. 20.000	

Desidero ricevere i libri indicati nella tabella, a mezzo pacco postale, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE. - SI RICHIEDE L'EMISSIONE DI FATTURA

PARTITA IVA

PAGAMENTO:

- Anticipato, mediante assegno bancario o vaglia postale per l'importo totale dell'ordinazione.
- Contro assegno, al postino l'importo totale.

AGGIUNGERE L. 3000 per contributo fisso spedizione. I prezzi sono comprensivi di I.V.A.

Sincronia

di **D.O. Wright**
trad. e adatt.
di **Angela Spinazzola**



Il computer si presta magnificamente come controllore dei nostri riflessi e penso di poter invitare a scagliare la prima pietra il lettore che non si è mai cimentato con un programma ad hoc. Ad ogni modo, se proprio vi fosse qualcuno sfuggito alla prova, ecco pronto lo Spectrum a rimediare. Schivo la prima pietra e passo a presentare l'oggetto del gioco. Sullo schermo compare una croce e a quattro bracci e dall'estremità di questi si muovono verso il centro altrettanti caratteri grafici uguali. Al centro dei bracci si sussegue invece una serie di 10 disegni, uno dei quali sarà uguale a quelli precedentemente citati. Quando vedrete in tutte le posizioni lo stesso disegno dovrete essere svelti a premere lo zero. Vi verrà accreditato il punteggio e il gioco continuerà con un'altra prova. È possibile scegliere tra 5 livelli di difficoltà, che si differenziano per il tempo lasciatovi a disposizione per premere lo zero. Se godete di salde meningi, cercate di memorizzare l'ordine con cui i disegni appaiono (sono contenuti nella stringa a\$), dopo di che potrete influire sull'ordine di lettura attraverso i tasti 5 e 8. Un punteggio appena decente deve essere superiore a 1000. Io non ci sono arrivata, ma non è detto che dobbiate avere i centri nervosi appannati quanto i miei.

Descrizione del programma

Linee	Commento
70	matrice contenente le istruzioni
90	livello di gioco
150	x è il numero del disegno che verrà stampato al centro s è il punteggio raggiunto h sono i colpi effettuati l sono le prove a disposizione per ogni partita
170-230	preparazione dello schermo
290	b è il disegno che verrà stampato sui bracci
310	stampa i disegni nei bracci
330	cancella i disegni nei bracci
360	sceglie il disegno nella stringa a\$...
400	...e lo stampa al centro
520-530	stampa il punteggio raggiunto
600-630	caricamento UDG
730-920	riempimento matrice con le istruzioni...
930-950	...e ciclo di stampa delle stesse

Sincronia versione per ZX Spectrum 16K/48K

```
10 INK 0: BRIGHT 1: CLS
15 GO SUB 600
25 LET a$="ABCDEFHJG"
27 PRINT PAPER 6;"A";a$a$a$a;"A"
```

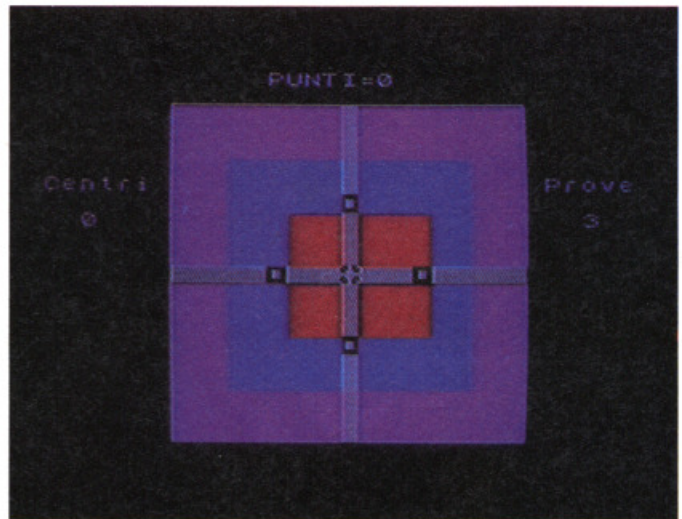
```
29 PRINT ' PAPER 7;"-> OCCHIO AL
    SINCRONISMO <-"
30 PRINT ' PAPER 6;"A";a$a;a$a;"A"

35 PRINT ' PAPER 5;" ISTRUZI
    ONI? (S/N) "
38 PAUSE 0
40 IF INKEY$="n" THEN GO TO 79
```

```

70 DIM x$(20,32): GO SUB 730
75 PRINT PAPER 5; INK 0;"Premi un
tasto per incominciare.":
PAUSE 0
79 INK 7
80 CLS : PRINT AT 10,4;"Livello di
gioco (1/5)"
90 LET q$=INKEY$
100 IF CODE q$<49 OR CODE q$>53
THEN GO TO 90
110 PRINT FLASH 1;AT 12,16;q$
120 BEEP 3,0
130 LET sl=VAL q$
140 CLS
150 LET x=INT (RND*10)+1: LET s=0:
LET h=0: LET l=3
170 PLOT 55,7: DRAW 0,153: DRAW 153,
0: DRAW 0,-153: DRAW -153,0
180 FOR i=2 TO 20: PRINT AT i,7;
INK 3;"{19SG8}": NEXT i
190 FOR i=5 TO 17: PRINT AT i,10;
INK 1;"{13SG8}": NEXT i
200 FOR i=8 TO 14: PRINT AT i,13;
INK 2;"{8SG8}": NEXT i
210 PRINT AT 11,7; INK 7;"{19SG8}"
220 FOR i=2 TO 20: PRINT AT i,16;
INK 7;"{SG8}": NEXT i
230 PRINT AT 11,16; FLASH 1; PAPER 2
;"{SG8}"
240 PRINT AT 0,12;"PUNTI=";s
250 PRINT AT 6,0;"Centri";AT 8,2;h
260 PRINT AT 6,27;"Prove";AT 8,29;l
270 LET i=1
290 LET b=INT (RND*10)+1
300 LET i=i+1: IF i=12 THEN GO TO 4
20
310 PRINT AT 11,27-i;a$(b);AT 11,5+i
;a$(b);AT i,16;a$(b);AT i-22,16;
a$(b)
320 BEEP .02,i
330 FOR q=1 TO (sl*10): NEXT q
340 PRINT AT 11,27-i; PAPER 7;" ";
AT 11,5+i; PAPER 7;" ";AT i,16;
PAPER 7;" ";AT i-22,16; PAPER 7
;" "
350 LET z$=INKEY$
360 LET x=x+(z$="8")-(z$="5")
370 IF z$="0" AND x=b THEN GO TO 47
0
380 IF x<1 THEN LET x=10
390 IF x>10 THEN LET x=1
400 PRINT AT 11,16;a$(x TO x)
410 GO TO 300
420 BEEP .5,-30
430 IF i=12 THEN LET l=l-1
440 PRINT AT 8,29;l
450 IF l=0 THEN PRINT AT 11,11;
FLASH 1;"fine partita": BEEP 1,
-10: BEEP 1,-20: BEEP 1,-30:
PRINT AT 11,7; FLASH 1;"Giochi
ancora?(S/N)": GO TO 550
460 GO TO 270
470 IF x=b THEN LET s=s+(b*10)
480 IF x=b THEN LET h=h+1
490 IF x=b THEN BEEP .1,20

```



```

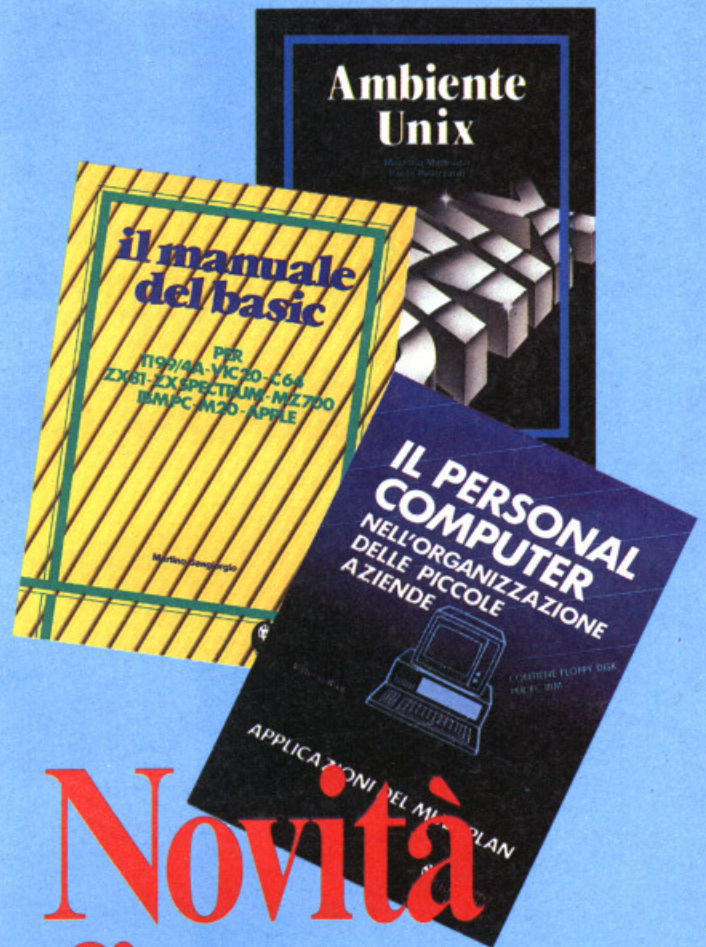
500 IF sl=1 THEN LET sl=.1
510 IF h=25 OR h=50 OR h=75 OR h=100
OR h=125 THEN LET sl=sl-1
520 PRINT AT 0,18;s
530 PRINT AT 8,2;h
540 GO TO 270
550 LET z$=INKEY$
560 IF z$="" THEN GO TO 550
570 IF z$="s" THEN FOR i=1 TO 10:
BEEP .1,i: NEXT i: GO TO 80
580 IF z$="n" THEN FOR i=10 TO 1
STEP -1: BEEP .1,i: NEXT i:
PRINT AT 11,0;"ARRIVEDERCI ALLA
PROSSIMA VOLTA!": BEEP .1,i:
STOP
590 GO TO 550
600 RESTORE 630
610 FOR a=USR "a" TO USR "k"-1:
READ c: POKE a,c: NEXT a
620 RETURN
630 DATA 255,255,195,195,195,195,255
,255
640 DATA 255,129,189,165,165,189,129
,255
650 DATA 0,0,60,60,60,60,0,0
660 DATA 231,195,129,24,24,129,195,2
31
670 DATA 153,90,60,255,255,60,90,153
680 DATA 129,66,60,60,60,60,66,12
9
690 DATA 255,231,231,129,129,231,231
,255
700 DATA 0,24,24,126,126,24,24,0
710 DATA 195,129,0,0,0,0,129,195,231
,219,165,90,90,165,219,231
730 LET x$(1)="Sei chiamato a sincro
nizzare il"
760 LET x$(2)="centro della figura c
on i 4 "
770 LET x$(3)="simboli che appaiono
ai lati e "
780 LET x$(4)="che si avvicinano.
790 LET x$(5)="Usa i tasti ~5~ ed ~8
~ per "

```

LOAD

Sincronia

```
800 LET x$(6)="cambiare il simbolo d
i centro "
810 LET x$(7)="Ottenuta la sincronia
premi ~0~."
820 LET x$(8)="Livello di abilita':"
830 LET x$(9)="da 1 (difficile)... a
5 (facile)"
840 LET x$(10)="{G3} {SG3} {G3}
{SG3} {G3} {SG3} {G3} {SG3}
{G3} {SG3} {G3} {SG3} {G3}
{SG3} {G3} {SG3}"
850 LET x$(11)="Ogni 25 centri il gi
oco accelera"
860 LET x$(12)="B U O N D I V E R T
I M E N T O"
8000 PRINT : FOR n=1 TO 13: FOR i=1
TO 32: PRINT PAPER 6;x$(n,i
TO i);: NEXT i: BEEP 1,n:
NEXT n
8100 RETURN
8500 STOP
9000 BORDER 0: PAPER 0: INK 6: CLS :
PRINT FLASH 1;AT 10,0;"{32G3}
FERMA IL NASTRO
{32SG3}"
9010 PRINT AT 21,0;"Premi un tasto pe
r incominciare.": PAUSE 0: RUN
9990 SAVE "sincronia" LINE 9000
9995 VERIFY "": STOP
```



Novità firmate Jackson.

IL PERSONAL COMPUTER
NELL'ORGANIZZAZIONE DELLE PICCOLE
AZIENDE: APPLICAZIONI DEL MULTIPLAN
COD. 578P L. 29.000

IL MANUALE DEL BASIC
COD. 534A L. 45.000

AMBIENTE UNIX
COD. 543P L. 19.000



GRUPPO
EDITORIALE
JACKSON

La biblioteca che fa testo.

Ritorna in edicola

VIDEO BASIC

Il corso più entusiasmante su cassetta
del Gruppo Editoriale Jackson per Commodore 64,
VIC 20 e Spectrum

200.000 copie vendute

del 1° fascicolo della prima edizione

Ogni lezione
uno spettacolo

Col 1° fascicolo
una cassetta giochi



Il corso è composto da:
20 fascicoli + (Quattordicinali)
20 cassette +
5 splendidi raccoglitori

Oggi è davvero facile imparare il Basic. Con Video Basic il corso su cassetta che ti permette di programmare subito il tuo computer. È facile: tu chiedi, lui risponde, tu impari. Passo dopo passo. Sul tuo schermo appaiono le domande, le risposte, gli esercizi e

tu, senza fatica, presto e bene, impari a conoscere e programmare il tuo computer, sia esso un VIC 20, un Commodore 64 o un Sinclair.

Video Basic è in edicola.
Provalo subito.

Ogni lezione è uno spettacolo.

Oggi il Basic si impara così. Video Basic, il corso su cassetta per parlare subito col tuo computer.

Video Basic
per imparare non solo il Basic.



Un'altra grande idea firmata
GRUPPO EDITORIALE JACKSON

Milano-San Francisco-Londra-Madrid

Gli ultimi saranno i primi?

Giudicalo tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione.

Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L.
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. Cod.
Cod. Cod.
Cod. Cod.

- Contanti allegati
- Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano
- Pagherò in contrassegno al postino al ricevimento dei volumi

Nome

Cognome

Via

CAP Città Prov.

Se richiesta fattura - codice fiscale

Data

J. soft: libri di qualità

Per Commodore 64

Giochi fantastici per il C 64
Cod. ASOC 001 L. 42.000
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vantaggioso per l'apprendimento delle tecniche di programmazione.

Il libro del C 64
Volume 1
Cod. ASOC 010 L. 24.000
(inclusa cassetta)
Il libro del C 64
Volume 2
Cod. ASOC 011 L. 24.000
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.



C 64: suono e grafica - Volume 1
Cod. ASOC 008 L. 24.000
(inclusa cassetta)
C 64: suono e grafica - Volume 2
Cod. ASOC 009 L. 24.000
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.

Per ZX Spectrum

15 giochi grafici per ZX Spectrum
Cod. ASOC 002 L. 24.000
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti

all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.



Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.

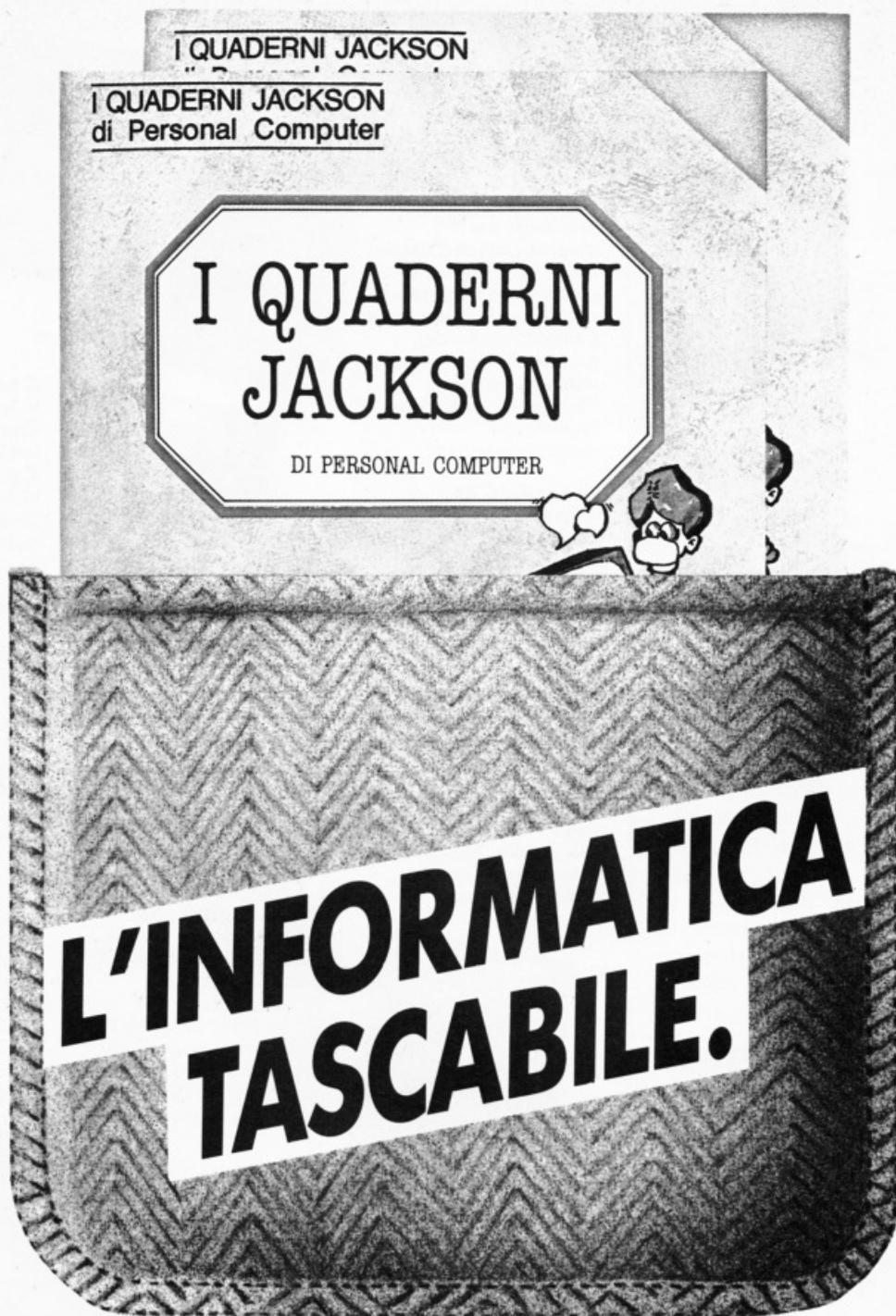
Paper Book - programmi per Apple II -
Cod. ASOC 005 L. 18.000
Paper Book - programmi per VIC 20 -
Cod. ASOC 003 L. 18.000
Paper Book - programmi per Commodore 64 -
Cod. ASOC 004 L. 18.000
Paper Book - programmi per Texas TI 99/4A -
Cod. ASOC 007 L. 18.000
Paper Book - programmi per ZX Spectrum -
Cod. ASOC 006 L. 18.000



In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, adventure una miscela esplosiva di software "pronto all'uso"!

J. soft s.r.l.

Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843



Arrivano i Quaderni Jackson, tanti volumi monografici per conoscere bene il personal computer e l'informatica.

Nei quaderni Jackson c'è tutto quello che è importante sapere sui computer, la programmazione, i linguaggi, il software, le applicazioni e i nuovi sviluppi dell'informatica.

Quaderni Jackson: l'informatica a tutti i livelli, in una collana aperta, pratica, essenziale, aggiornata.

L'informatica tascabile per chi vuole saperne di più e compiere così un salto di qualità nel mondo di oggi e di domani.

Ogni mese, 2 volumi.

Volumi già pubblicati:

Gianni Giaccaglioni "Vivere col Personal Computer"

Paolo Bozzola "Dentro e fuori la scatola"

Enrico Odetti "Ed è subito BASIC Vol. I"

Enrico Odetti "Ed è subito BASIC Vol. II"

Paolo Capobussi

e Marco Giacobazzi "A ciascuno il suo Personal"

Fulvio Francesconi

e Fernando Paterlini "To do or not to do"

Il castello stregato

di Babar Abidi
trad. e adatt.
di Paolo Maffei

Riuscirete a sfuggire ai fantasmi che abitano il castello stregato?

Siete alla ricerca di tre preziose anfore nascoste in un solaio del castello incantato e questo non creerebbe troppi problemi, se non vi fossero a custodire le preziose anfore nientemeno che spettri velocissimi e terribili scheletri.

Per raggiungere il solaio dovrete utilizzare le numerose scale che troverete nel castello, cercando di trovare l'itinerario migliore per evitare di essere catturati dal fantasma.

Durante il cammino troverete anche un martello, che vi servirà per eliminare lo scheletro che si trova all'ultimo piano. Per combattere il fantasma, invece, avete a disposizione un numero illimitato di trappole, che potrete depositare premendo il tasto z (ricordatevi che questa sarà la vostra arma migliore!). Una volta raggiunto il solaio, e presa un'anfora, dovrete uscire dal castello, mettere il prezioso carico nel vostro furgoncino e tornare a recuperare le altre due.

Per muovervi utilizzate i seguenti tasti: q per salire, a per scendere, o per andare a sinistra, p per andare a destra, z per depositare una trappola.

Il programma contiene una semplice routine in linguaggio macchina (linee 9990-9993), per ottenere speciali effetti sonori, accessibile tramite RANDOMIZE USR 35000; questa routine potrà essere utilizzata anche in altri vostri programmi.



Il castello stregato versione per ZX Spectrum 48K

```

4 CLEAR 34999: LET hs=0: LET h$="A
  JM": PRINT AT 10,5; FLASH 1;"Att
  endere, prego.": GO TO 1000

5 IF ATTR (a+1,b)=6 THEN PRINT
  AT a+1,b; INK 4;"M"

50 PRINT OVER 1; INK 2; PAPER 8;
  AT a,b;"A"
  
```

```

51 IF a=4 AND b=15 AND h=1 THEN
  BEEP .1,10: LET t=t+5: IF t>=20
  THEN LET t=20: PRINT AT 2,0;"
  ": LET h=0: LET s=s+100: PRINT
  AT 3,15; INK 6;" "

52 LET a=a+(INKEY$="a" AND ATTR (a+
  1,b)=2)-(INKEY$="q" AND ATTR (a-
  1,b)=2)

53 IF a=12 AND b=26 AND ATTR (a,b+1
  )=5 THEN BEEP .1,20: PRINT AT 1
  2,27; INK 7;" ": PRINT AT 2,0;"G
  ": LET h=1
  
```

```

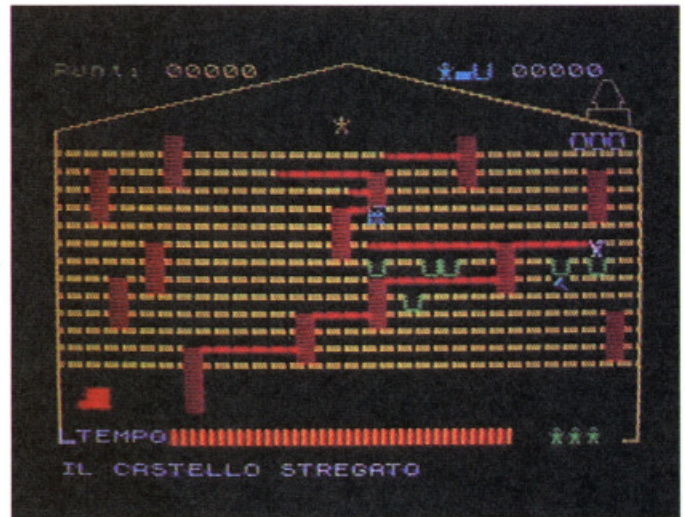
54 LET b=b+(INKEY$="p" AND b<=29
  AND ATTR (a,b+1)<>5)-(INKEY$=
  "o" AND b>=2 AND ATTR (a,b-1)
  <>5)
55 PRINT OVER 1;AT a,b; INK 9;"A"
56 IF INKEY$="z" THEN GO SUB 5
57 PRINT AT 0,11-LEN STR$ s; INK 6;
  s
59 LET t=t-.015: IF t<=0 THEN
  GO SUB 800
60 PRINT AT 21,6; INK 6; PAPER 2;t$
  ( TO t); PAPER 0;" "
62 PRINT AT c,d; OVER 1; INK 2;"C"
63 IF ATTR (c+1,d)=4 THEN POKE 350
  06,5: POKE 35025,28: RANDOMIZE
  USR 35000: PRINT AT c+1,d; INK 5
  ; BRIGHT 1;"N": POKE 35025,29:
  RANDOMIZE USR 35000: PRINT AT c
  +1,d; INK 6;"E": LET c=4:
  RANDOMIZE : LET d=(RND*10)+1:
  LET s=s+25
64 LET d=d+(d<=b AND ATTR (c,d+1)
  <>6)-(d>=b)
65 LET c=c-(c>=a AND ATTR (c-1,d)
  <>6 AND ATTR (c-1,d)<>4)+(c
  <=a AND ATTR (c+1,d)<>6 AND
  ATTR (c+1,d)<>4)
69 PRINT OVER 1;AT c,d; INK 5;"C"
80 IF a=4 AND b=26 AND g=0 THEN
  LET j=j-1: LET g=1: PRINT AT 4,
  28; INK 7;j$( TO j);" ": BEEP .1
  ,50: LET u=1
85 IF a=19 AND b=3 AND u=1 THEN
  LET s=s+275: POKE 35056,5:
  RANDOMIZE USR 35000: LET u=0:
  LET g=0: IF j=0 THEN GO TO 200

99 IF a=INT c AND b=INT d THEN
  GO TO 800
100 GO TO 50
200 LET s=s+INT (t*50)
210 PRINT AT a,b;" ";AT c,d;" "
215 PRINT AT 20,1; BRIGHT 1; FLASH 1
  ;"
  B O N U S
  "
220 POKE 35006,100: RANDOMIZE USR 35
  000
240 LET a=19: LET b=3: LET c=4:
  LET d=26: LET h=h+1: LET j=3
250 LET t=20-h: IF t<=10 THEN LET t
  =10
260 PRINT OVER 1;AT c,d;"C"
270 PRINT AT 4,28;j$( TO j)
280 PRINT AT 20,1; FLASH 0;"
  "
290 GO TO 51
800 PRINT AT a,b; OVER 1;"A";AT c,d;
  OVER 1;"C"
801 PRINT AT 21,27;" "
810 FOR n=10 TO 2 STEP -2: BEEP .1,n
  : PAUSE 1: NEXT n
815 PAUSE 7: BEEP .075,8: PAUSE 1:
  PRINT AT a,b;"N": BEEP .075,8
820 PRINT AT a,b;" "
831 LET m=m-1: IF m<=0 THEN GO TO 9
  00

```



**Il castello
stregato**



```

840 PRINT AT 21,28; INK 4;m$( TO m);
  " "
850 LET a=19: LET b=3: LET c=4:
  LET d=26: LET g=0: PRINT AT 4,2
  8;j$( TO j)
860 PRINT AT c,d; OVER 1;"C"
890 GO TO 51
900 IF s>=hs THEN LET hs=s: PRINT
  AT 5,0; INK 7; PAPER 2;"NUOVO RE
  CORD!!""INTRODUCI LE TUE INIZIA
  LI"
901 IF s=hs THEN FOR n=-50 TO 50
  STEP 5: BEEP .1,n: NEXT n
902 IF s=hs THEN INPUT h$: IF LEN h
  $>3 THEN GO TO 902
903 PRINT AT 21,28;" "
931 PRINT AT 21,27;" ": PRINT AT 11,
  0; INK 6; PAPER 1; FLASH 1;"
  GAME OVER
  "
932 PRINT AT 15,0; INK 0; PAPER 5;"
  VUOI GIOCARE ANCORA? S/N "
933 IF INKEY$="s" THEN RESTORE 9927
  : GO SUB 9900: GO SUB 9920:
  GO TO 51
934 IF INKEY$="n" THEN STOP
940 GO TO 920
1000 GO SUB 9990
1010 GO SUB 9900
1020 GO SUB 9800
1030 GO SUB 9700
1040 GO SUB 9920
1050 GO TO 51
9700 BORDER 1: PAPER 1: INK 7: CLS
9710 LET i$="Benvenuti nel CASTELLO S
  TREGATO.Dovrete correre nel ca
  stello cercando di recuperare l
  e giare poste sotto il camino.Og
  ni giararecuperata va portata ne
  l vostro furgoncino."

```



```

9715 LET i$=i$+" Nel vostro cammino
      incontrerete.. C il Fantasma
                    E lo Scheletro
      Il Fantasma puo' essere
      combattuto lasciando delle
      trappole, lo Scheletro puo' essere
      colpito con il martello
      che troverete nel castello."
9718 LET i$=i$+"
      ~Q~ Su
      ~A~ Giu' ~O~ Sinistra
      ~P~ Destra ~Z~ Lascia una t
      rappola"
9719 PRINT INK 6; FLASH 1;" IL C
      ASTELLO STREGATO "
9720 PRINT : FOR n=1 TO LEN i$:
      BEEP .005,5: PRINT i$(n);:
      NEXT n
9725 POKE 35004,255: POKE 35006,50:
      RANDOMIZE USR 35000:
9730 PRINT #0;" premi un tasto."
9740 PAUSE 0
9750 POKE 35025,29: RANDOMIZE USR 350
      00:
9760 RETURN
9803 FOR y=65368 TO 65479
9810 READ z: POKE y,z: NEXT y
9820 DATA 56,40,146,124,56,56,40,108,
      129,255,129,255,129,255,129,255
9830 DATA 62,127,73,73,127,127,85,85,
      127,34,65,65,65,65,65,62
9840 DATA 8,93,73,62,8,20,34,34,247,2
      47,247,0,0,0,0,0
9850 DATA 0,24,56,112,104,4,2,0,51,51
      ,51,51,51,51,51,51
9860 DATA 0,0,0,0,15,7,7,31,0,0,0,0,2
      55,254,254,254,39,71,127,127,127
      ,255,40,16,254,254,254,254,254,2
      55,20,8
9870 DATA 129,195,129,195,129,195,129
      ,126,0,73,42,0,99,0,42,73
9890 RETURN
9900 LET t=20: LET t$="HHHHHHHHHHHHHHH
      HHHHHHHHHHHHHHHH"
9902 LET j=3: LET j$="DDD"
9903 LET a=19: LET b=3: LET c=4:
      LET d=26
9904 LET m$="AAA": LET m=3
9905 LET u=0: LET g=0
9907 LET h=0
9910 LET s=0
9918 RETURN
9920 BORDER 0: PAPER 0: INK 7: CLS :
      FOR n=5 TO 17: PRINT AT n,0;
      INK 6;"FFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFF": NEXT n
9925 FOR n=1 TO 14: READ y,x: PRINT
      AT y,x; INK 2;"B";AT y+1,x;
      INK 2;"B";AT y-1,x; INK 2;"B":
      NEXT n
9927 DATA 5,6,5,22,7,2,7,17,7,29,9,15
      ,11,5,11,24,13,3,13,17,15,13,15,
      30,17,7,18,7
9930 FOR n=1 TO 11: READ y,x: DRAW
      INK 6;y,x: NEXT n

```

```

9932 DATA 0,144,127,31,1,0,127,-31,0,
      -144,-254,0,0,144,127,31,126,-32
      ,0,-142,-254,0
9933 PLOT 0,0
9934 PLOT 250,146: FOR n=1 TO 3:
      READ x,y: DRAW INK 6;x,y:
      NEXT n
9935 DATA 0,8,-18,0,0,-4
9937 PLOT 246,154: FOR n=1 TO 7:
      READ x,y: DRAW INK 6;x,y:
      NEXT n
9939 DATA 0,3,2,0,-5,10,-4,0,-5,-10,2
      ,0,0,-3
9940 PRINT AT 18,1; INK 2;"IJ";AT 19,
      1; INK 2;"KL"
9942 PRINT AT 12,27; INK 5;"G"
9945 PRINT AT 4,28;j$( TO j)
9947 PRINT AT 21,1; INK 7;"TEMPO";
      INK 6; PAPER 2;t$( TO t);
      PAPER 0;" "; INK 4;m$( TO m)
9949 PRINT AT 21,30;" "
9950 PRINT #0; INK 7;"IL CASTELLO STR
      EGATO "
9952 PRINT AT 0,0; INK 4;"Punti ";
      INK 6;"00000";AT 0,21; INK 5;h$
      ; INK 7;" 00000"
9953 PRINT AT 3,0; OVER 1; INK 6;" "
9954 PRINT AT 4,26; OVER 1; INK 2;"C"
9955 PRINT AT 0,29-LEN STR$ hs; INK 7
      ;hs
9960 PRINT AT 3,15; INK 6; PAPER 0;"E
      "
9970 PRINT ; INK 6;AT 18,6;" ";AT 18,
      8; INK 6;" "
9980 RETURN
9990 DEF FN a(n)=(CODE a$(n)-48 AND
      CODE a$(n)<58)+(CODE a$(n)-55
      AND CODE a$(n)>64)
9991 LET a$="F31110D0260A3A485C1F1F1F
      0EFEEEl0ED794310FE2520F41C1520E8
      FBC9"
9992 FOR J=1 TO LEN A$ STEP 2
9993 POKE 35000+J/2,16*FN A(J)+FN A(J
      +1): NEXT J
9994 RETURN
9995 REM A B C D E F G H I J K
9996 REM A B C D E F G H I J K
9997 REM
9998 REM L M N
9999 REM L M N

```



Specpaint

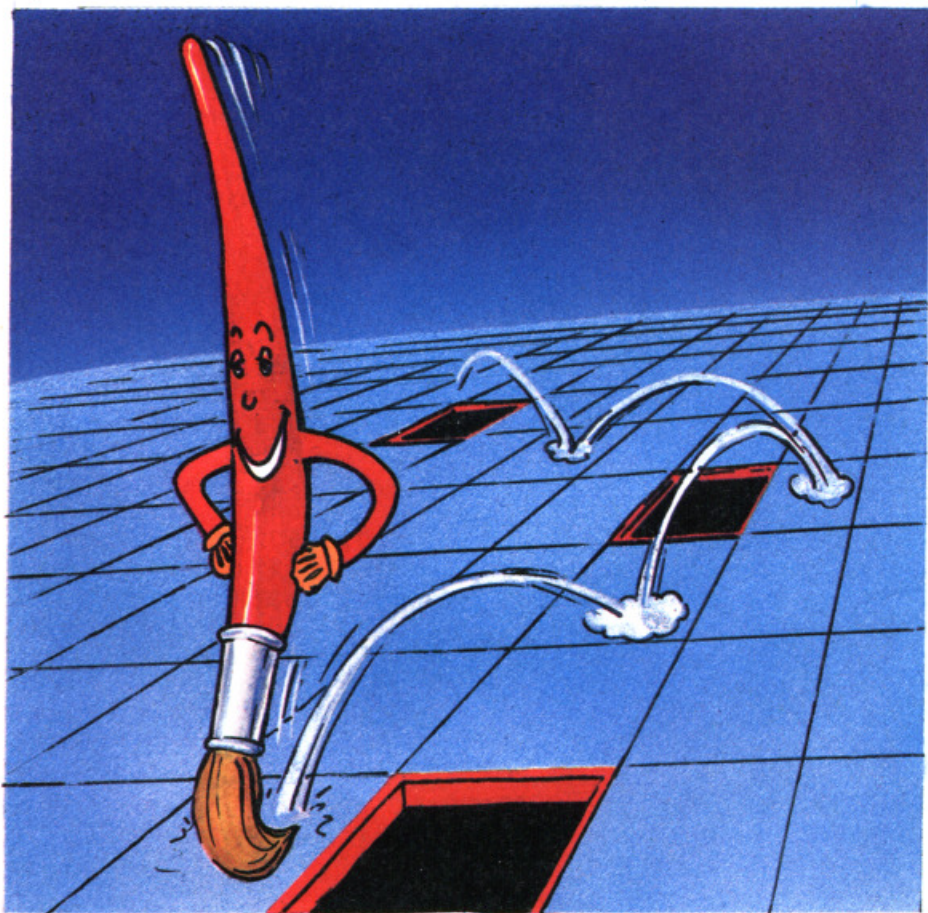
di **Carlo Panzalis**

Un coloratissimo gioco di azione, riflessi e strategia!

I lettori che hanno avuto occasione di vedere all'opera Macintosh, ultima creatura della Apple, resteranno forse un po' delusi scoprendo che Specpaint, pur richiamando subito alla mente Mac Paint, non costituisce una trasposizione sullo Spectrum del fantastico programma di Bill Atkinson! Il programma qui presentato è invece un gioco che si avvale di simboli grafici ispirati agli strumenti di lavoro di Mac Paint (il pennello, i barattoli di vernice, ecc.) e che si propone tuttavia, come obiettivo finale, di mostrare al lettore come strutturare un programma, come realizzare e muovere caratteri grafici di grandi dimensioni, come sfruttare operativamente la variabile di sistema UDG per aumentare il numero di caratteri grafici definiti, ed altro ancora.

Iniziamo dunque subito a chiarire lo scopo del gioco, per passare poi ad esaminare le caratteristiche del listato. Chi è interessato ad approfondire gli aspetti della programmazione BASIC dello Spectrum potrà così trovare chiarimenti e utili suggerimenti per rendere i propri listati più ordinati e i programmi più efficienti.

Elementi essenziali del gioco sono una scacchiera di sei quadrati di lato, cinque barattoli di vernice posti a destra nello schermo e una grossa freccia blu posta a destra della scacchiera. La freccia costituisce l'interfaccia fra scacchiera e barattoli, nel senso che solo dalla freccia è possibile intingere il pennello nei barattoli di vernice per poi rientrare sulla scacchiera, pronti a colorarla come il gioco richiede. Il giocatore dovrà, in una prima fase, colorare le sei linee orizzontali in base alle specifiche di sei indici per il colore, posti a sinistra della scacchiera stessa. Sarà dunque necessario intingere il pennello in uno dei quattro barattoli di vernice previsti, ciascuno dei quali è associato a uno dei primi quattro tasti numerici (magenta-1, verde-2, azzurro-3, giallo-4), e dovrà poi spostarsi entro la scacchiera per verni-



Descrizione del programma

Linee	Commento
1 - 90	inizializzazione
100 - 390	avvio gioco, inizializzazione
400 - 1000	loop principale
2000 - 2850	routine per il dimostrativo (istruzioni)
3000 - 3500	memorizzazione e ordinamento migliori punteggi
4000 - 4900	comunicazione della corretta verniciatura della scacchiera
5000 - 5550	apparizione delle tessere rosse
7000 - 7777	uscita sulla freccia
8000 - 8820	allestimento del display
9000 - 9454	caratteri grafici
9500 - 9840	istruzioni
9900 - 9999	routine di servizio (sonoro, demo istruzioni...)

ZX Spectrum Expansion System

L'alternativa della Sinclair ai floppy disc

Lo ZX Spectrum Expansion System contiene:

- **Uno ZX Microdrive** - Che amplia la possibilità dello ZX Spectrum in quei settori, come quelli della didattica e delle piccole applicazioni gestionali, dove è necessaria una veloce ricerca delle informazioni memorizzate su un supporto magnetico.
- **Una ZX Interface 1** - Indispensabile per il collegamento dello ZX Microdrive. Incorpora una interfaccia RS 232 e un sistema di collegamento in rete locale.
- **Quattro cartucce Microdrive comprendenti un programma di:**
 - Word processor «Tasword Secondo»
 - Masterfile filing system
 - Inventore di giochi
 - Le formiche giganti
- Un programma dimostrativo del Microdrive
- Documentazione per il collegamento, il funzionamento e altre descrizioni tecniche.
- Cavi di collegamento allo ZX NET che può collegare fino a 64 computer ZX Spectrum o QL.



In vendita presso
i rivenditori specializzati

ciare la linea desiderata. Per colorare una linea di diverso colore il giocatore dovrà tornare sulla freccia blu, cambiare colore e tornare sulla scacchiera. Una volta verniciate le sei linee orizzontali, come richiesto, verrà composta sullo schermo una nuova scacchiera: questa volta lo strumento a disposizione sarà un classico rullo da imbianchino, col quale il giocatore dovrà tentare di verniciare le sei linee verticali, sempre secondo quanto impongono gli indici per il colore, questa volta posti al di sotto della scacchiera. Al termine di questa seconda fase il giocatore conoscerà il tempo da lui impiegato a raggiungere lo scopo e il punteggio conseguito.

Una caratteristica essenziale del gioco consiste nel fatto che il pennello o il rullo, una volta selezionato un colore, "imbrattano" con quella tinta tutte le tessere su cui passano. È dunque opportuno che il giocatore elabori una strategia di movimento, onde evitare di trovarsi nella situazione dell'omino della famosa barzelletta che dopo aver verniciato accuratamente il pavimento di una stanza si rende conto che la porta si trova nella parete opposta.

È fondamentale ricordarsi che la routine atta a verificare la corretta verniciatura della scacchiera viene, in ogni fase, attivata dall'uscita sulla freccia. Eseguito dunque il compito assegnatovi, posizionate il pennello/rullo sulla grossa freccia e attendete qualche istante: se tutto è stato colorato come richiesto, accederete alla fase successiva oppure otterrete le informazioni circa la qualità della vostra performance, naturalmente se avete già ultimato la prima fase.

Per ostacolare il giocatore, e per rendere più interessante il gioco, è stata prevista l'apparizione casuale di tessere rosse non eliminabili con le normali vernici, ma solo con una energica mano di vernice bianca (quinto barattolo associato al tasto 5), e che impediscono di vincere la partita rapidamente e senza sforzo. Notate che il bianco non ha altra funzione di quella ora vista: questo colore non viene mai richiesto dagli indici per il colore per verniciare una linea o una colonna. È evidente che ogni volta che entreremo sulla scacchiera, dopo aver selezionato la vernice bianca per cancellare una tessera rossa, sporcheremo di bianco anche le tessere che avevamo precedentemente colorato correttamente e che dovremo dunque successivamente riverniciare. Il programma utilizza la variabile di sistema FRAMES per il corretto conteggio dei secondi, prevede quattro distinti livelli di difficoltà ed è dotato di una routine per la memorizzazione e l'ordinamento dei migliori punteggi. Inoltre il gioco è corredato di istruzioni auto-esplicative, nel senso che il programma stesso tenta di mostrare, tramite un'animazione, gli scopi del gioco ed il movimento del pennello/rullo.

Vincere una partita non è facile, in special modo selezionando i livelli di diffi-

coltà più elevati. La caratteristica essenziale del gioco è comunque costituita da un uso del colore che non è, come solitamente avviene, semplice elemento decorativo, ma elemento centrale di ogni partita.

Ricordiamo qui brevemente i comandi, costituiti dai tasti "A" & "Z" per i movimenti verticali; "J" & "K" per i movimenti orizzontali; da "1" a "5" per selezionare le vernici; infine il tasto "0", attivabile solo dopo aver posizionato il pennello o il rullo sulla freccia, con il quale si ottiene un arresto temporaneo del gioco: seguendo le indicazioni del programma si potrà così uscire dal gioco, iniziare una nuova partita (eventualmente scegliendo un nuovo livello di difficoltà) o continuare la partita come se l'interruzione non fosse avvenuta.

Analizziamo il listato

Conclusa questa panoramica sul gioco vero e proprio, passiamo ora a considerare le peculiarità del listato: in tal senso ci soffermeremo soprattutto sugli aspetti che possono fornire utili indicazioni che il lettore potrà trasferire nei propri lavori, allo scopo di migliorarne struttura e funzionamento.

Il programma, così come appare in queste pagine, gira sullo Spectrum 48K. È tuttavia possibile, attraverso un paio di artifici, renderlo compatibile con la versione a 16K. Poiché per quanto riguarda tale conversione i problemi da considerare sono molteplici, risulta più opportuno chiarire dapprima la logica del programma, per poi occuparsi della sua trasformazione. Per agevolare la comprensione del discorso, ogni qualvolta dovremo fare preciso riferimento a locazioni di memoria (ad esempio nel caso degli UDG), considereremo il caso dello Spectrum 48K: chi volesse adeguare i dati al 16K dovrà semplicemente sottrarre 32768 al valore da noi riportato per conoscere quali locazioni sono riservate a quel determinato scopo sulla sua macchina.

Struttura

Dall'esame del listato è agevole rilevare che il programma può essere diviso in una sezione principale (main program) e in una serie di subroutine, ciascuna delle quali assolve funzioni ben precise. Il programma è stato cioè strutturato in modo che, al verificarsi di date condizioni, si possa saltare dal loop principale ad una sezione del listato capace di far fronte alle mutate circostanze. In tal senso sono state predisposte le diverse subroutine, cui si accede dal main program o al limite da un'altra subroutine. Ogni singola condizione, ad esempio l'uscita sulla freccia per selezionare un nuovo colore o la verifica della possibile vittoria del giocatore, vengono specificate nell'ambito della appropriata subroutine in termini di istruzioni non elementari, cioè riconoscibili dal sistema

operativo e idonee a realizzare lo scopo per cui sono state create. Configurare un programma secondo tale tecnica presenta notevoli vantaggi, soprattutto sotto il profilo della chiarezza e leggibilità del listato, mentre consente al limite di operare su singole sezioni senza compromettere il funzionamento della struttura portante. Vedremo che proprio in questo caso tale aspetto è della massima importanza, per consentire la trasformazione del listato in modo da sfruttare il programma anche sullo Spectrum 16K.

Grafica

Per valorizzare Specpaint sono stati utilizzati ben 51 caratteri grafici: ciò è stato possibile operando sulla variabile di sistema UDG (locazioni 23675 & 23676), in modo da puntare tre distinte tavole di caratteri grafici, ciascuna delle quali viene "letta" in relazione alla differente fase di gioco considerata. Per non appesantire il discorso rinviemo quanti non fossero a conoscenza di questa tecnica ad un nostro precedente articolo (SUPERSINC n. 5; UDG 2ª Parte), mentre qui ci limiteremo ad indicare come sono stati ripartiti i simboli grafici fra le tre tavole. Nella prima tavola (quella abitualmente utilizzata), a partire dall'indirizzo 65368 e per 168 byte, vengono immagazzinati i simboli relativi all'allestimento dello schermo (barattoli di vernice, freccia, ecc.) e alla realizzazione del quadro informativo finale (omini, altri barattoli). Abbassando il puntatore di altri 168 byte, componiamo la seconda tavola di caratteri grafici (dall'indirizzo 65200), relativa alla prima fase del gioco e nella quale sono conservati il pennello e il suo "cancellino", vale a dire la maschera che va a spostarsi sulla posizione del perinello dopo ogni mossa, in modo da cancellarne l'immagine. Nella terza tavola, più bassa di altri 256 byte a partire dall'indirizzo 64944 fino a 65111, sono invece immagazzinati i simboli relativi alla seconda fase di gioco, cioè il rullo e il suo cancellino. Per evitare interferenze con lo stack, RAM-TOP viene abbassata di 424 byte, precisamente all'indirizzo 64943.

La necessità di utilizzare un numero così elevato di caratteri grafici definibili trova la sua ragione nella scelta di unire insieme diversi simboli, in modo da comporre immagini di grandi dimensioni. Così pennello, rullo, barattoli di vernice e tessere della scacchiera sono composti da una griglia di tre caratteri di lato (quindi da nove caratteri ciascuno). Le tre stringhe I\$, b\$ e u\$ contengono, rispettivamente, nella prima fase il pennello, il cancellino e il pennello in uscita sulla freccia; nella seconda fase il rullo, il cancellino e il rullo in uscita. A seconda della particolare fase del gioco (allestimento, prima fase, seconda fase) la variabile di sistema UDG punta la tavola richiesta in quel momento, realizzando così la coesistenza in una data fase

di immagini grafiche appartenenti a tavole differenti.

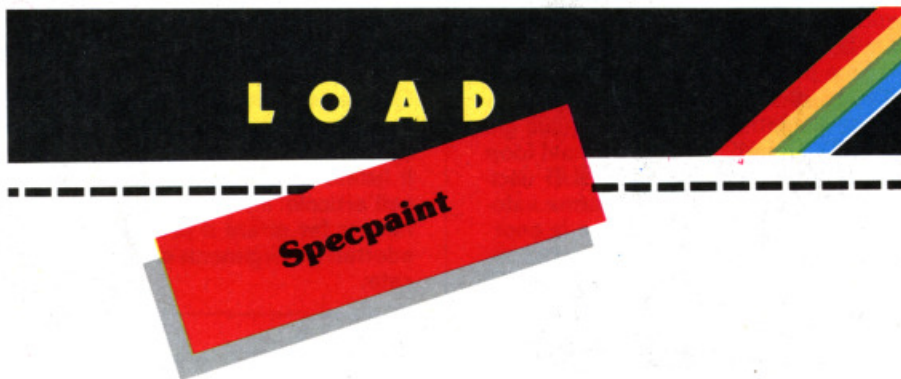
È qui anche opportuno sottolineare che la routine di caricamento degli UDG ed i dati ad essa relativi si presentano nel listato in modo da risultare il più possibile leggibili e decodificabili dall'utente, tutto sommato sacrificando a tale scopo la pura e semplice efficienza operativa. In altre parole si poteva realizzare una routine migliore per caricare gli UDG, ma probabilmente il lettore meno esperto avrebbe incontrato difficoltà maggiori nel comprenderne il funzionamento.

Accorgimenti di varia natura

Nella elaborazione di Specpaint sono stati utilizzati taluni accorgimenti che dovrebbero facilitare la lettura e la comprensione del programma, così che, al di là dell'aspetto giocoso, chi è interessato ai problemi della programmazione possa trarre il massimo vantaggio dall'esame del listato. Fra tali accorgimenti segnaliamo ad esempio i nomi assegnati alle variabili, la tecnica seguita nel predisporre le istruzioni, l'inserimento di sicure per gli INPUT, il meccanismo di verifica dell'eventuale vittoria, l'utilizzazione di FRAMES per il corretto conteggio dei secondi.

Per quanto riguarda il primo aspetto dall'esame del listato appare evidente che i nomi associati alle variabili sono idonei a consentire la rapida identificazione del compito svolto da ciascuna di esse (o quasi). Operare in questo modo è vantaggioso per almeno un duplice ordine di motivi, come già accennato precedentemente: innanzitutto perché si facilita notevolmente la lettura del programma all'utente che voglia comprendere come si possano realizzare determinati effetti; in secondo luogo perché in questo modo si facilita notevolmente il debug del programma. Come è noto, con tale termine, nel gergo dell'informatica, si intende la ricerca e l'eliminazione di un errore (bug = cimice, pulce), tipicamente non un errore di profondissima portata, ma comunque capace di disturbare la corretta esecuzione del programma. Quanto più il bug è piccolo, tanto più affannose saranno le ricerche da parte del povero programmatore. È chiaro che riconoscere un intoppo nello svolgimento del programma, e poterlo associare immediatamente ad una data variabile, consente di porre rapidamente rimedio all' indesiderata situazione. Purtroppo, associare alle variabili nomi estesi è, in termini di memoria, dispendioso, ed inoltre fa lievitare i tempi necessari a battere il listato; tuttavia, nel caso di programmi destinati a pubblicazione è opportuno non trascurare questo aspetto, ove è possibile.

Le istruzioni sono state realizzate in modo che, tramite una animazione, dimostrino scopi del gioco e relazioni fra tasti e movimenti. Inoltre, data la notevole quantità di caratteri grafici da caricare,



Descrizione delle variabili

ramtop	può assumere valore 127 (nel 16K) o 255 (nel 48K): serve ad indicizzare il programma in relazione alla RAM disponibile
costante, casseti	locazioni ove saranno conservati i valori degli attributi atti a consentire la verifica della eventuale vittoria
qui, la, flip, flop ma, mb, ms, md uscita	coordinate per la stampa del pennello/ruolo limiti estremi della scacchiera consente il salto alla subroutine che controlla l'uscita sulla freccia
sfondo, pap, inc	vari attributi atti a realizzare la scacchiera o a consentire la verniciatura
ruolo, salto cx, cy livello, diffi, coeff	parametri per salti condizionati coordinate tessere rosse variabili relative al livello di difficoltà prescelto
secondi, punti, max H\$(), X\$(), Y\$(), g\$ l\$, b\$, u\$, p\$	variabili relative al tempo e al punteggio utilizzate per la gestione dei migliori punteggi contengono simboli grafici (pennello, ruolo, ecc.) da stampare sullo schermo

Comandi

A & Z	alto & basso
J & K	sinistra & destra
1 - 5	selezione vernice (solo dalla freccia)
0	seleziona alcune opzioni (solo dalla freccia); in particolare, dopo averlo premuto si può terminare il gioco (premendo N), interrompere la partita in corso e iniziarne una nuova (premendo S), continuare la stessa partita (premendo C)

si è scelto di richiamare la routine di caricamento, mentre sullo schermo vengono mostrate le istruzioni, così che il lettore non si accorga (o quasi) dei tempi morti.

Nel listato sono state poi predisposte sicure in corrispondenza degli INPUT, cioè i dati che il giocatore deve inserire da tastiera per selezionare il livello di gioco, ecc., in modo che un errato inserimento non provochi l'arresto del programma. Si tratta di un accorgimento solitamente molto apprezzato da parte di chi deve pubblicare o commercializzare un programma, poiché in tal modo si evitano le accuse di mancato funzionamento del gioco, quando invece sono gli utenti che sbagliano ad usarlo. Realizzare programmi "a prova di errore" costa comunque molto in termini di memoria e di tempo, e spesso appesantisce la struttura del programma: un buon compromesso consiste nel predi-

porre sicure in caso di INPUT ambigui, che possono cioè indurre facilmente a sbagliare, e lasciare alla responsabilità dell'utente il controllo sulla correttezza di input che non possono dare addito ad equivoci; in altri termini, se l'utente risponde alla richiesta "Inserisci livello (1-3)" con un tasto letterale o con valori diversi da quanto chiaramente indicato, non può poi prendersela con chi ha elaborato il programma, ma deve arrabbiarsi con se stesso!

La routine che verifica la corretta verniciatura della scacchiera è richiamata ogni volta che il pennello o il ruolo vengono portati sulla freccia: se la colorazione non è stata ancora ultimata, la routine ne prende atto e lascia ancora il controllo nelle mani del giocatore; altrimenti essa effettua il salto alla fase successiva. In una prima elaborazione del programma detta routine era attivata solo su richiesta del giocatore, pre-

mando un determinato tasto; nello stabilire la struttura definitiva del gioco si è invece preferito operare come sopra riportato, onde evitare di aggiungere altri tasti di controllo accanto a quelli già numerosi previsti. In tal senso, nel loop principale, si trova un "GO SUB uscita": il parametro "uscita" assume valore 7000 (routine di verifica), se le coordinate del pennello sono in corrispondenza della freccia, oppure valore 7777 (RETURN) in caso contrario. In concreto la routine controlla la eguaglianza fra 36 attributi (le tessere della scacchiera) e i 6 valori (linee o colonne, rispettivamente se ci si trova nella prima o seconda fase) contenuti in altrettante locazioni di memoria, dette cassette (locazioni da 65150 a 65155). Il caricamento dei cassette avviene nella fase di allestimento del display: quando viene stampato un indice per il colore si carica contemporaneamente la corrispondente locazione di memoria con il valore dell'attributo relativo a quell'indice. Infine è stata utilizzata FRAMES (variabile di sistema fra 23672 e 23674) per calcolare il tempo impiegato a vincere, abbinando poi a tale variabile il livello di difficoltà, in modo da assegnare al giocatore differenti punteggi in relazione al diverso livello selezionato. Il tempo viene misurato dal momento in cui viene scelto il livello sino al termine della seconda fase.

Compatibilità 16K

Per concludere, qualche parola va spesa anche sugli accorgimenti volti a realizzare la compatibilità del gioco nelle due versioni dello Spectrum e quindi per consentirne il caricamento anche sul 16K. Poiché il programma richiede di abbassare RAMTOP e opera alterando frequentemente la variabile di sistema UDG per puntare la tavola di caratteri grafici richiesta in un dato momento, è stata approntata una procedura di "indicizzazione" del programma. L'indicizzazione consente cioè all'utente, nel realizzare la conversione, di non modificare i valori numerici che consentono di abbassare RAMTOP o che puntano una diversa area RAM, in modo da "leggere" altri caratteri grafici oltre ai 21 normalmente previsti. Ad adeguare automaticamente tali valori provvedono infatti le linee 10 e 20: la prima abbassa RAMTOP di 424 byte rispetto al valore di default, portando così il limite superiore dell'area BASIC a 32175 nel 16K e a 64943 nel 48K; la seconda linea attribuisce invece alla variabile RAMPTOP il valore 127 o 255, a seconda della versione considerata. In tal modo abbiamo "indicizzato" il programma rispetto alla RAM disponibile. I vantaggi derivanti da tali accorgimenti, spiace ammetterlo, vengono in parte vanificati dalla notevole lunghezza del programma (oltre 18K): ciò significa che, nonostante quanto detto, il lettore dovrà operare sul listato in modo da dividerlo

Avvertenze relative al programma versione 16K

Questa versione (su cassetta) gira solo sul 16K: chi possiede il 48K può fare a meno di caricarlo (a meno che si voglia resettare RAMTOP e la variabile di sistema UDG come sul 16K!).

Il listato di questa versione non è rigorosamente fedele a quanto descritto nell'articolo: infatti, per conseguire il rilevante risparmio di memoria, necessario per far girare il programma entro gli otto Kilobyte disponibili, sono stati adottati accorgimenti che lo differenziano parzialmente dalla versione pubblicata.

in sezioni, che possano essere caricate separatamente da nastro. In pratica Spepaint può essere diviso agevolmente in tre sezioni: istruzioni, set grafico, main program.

Chi volesse far girare il programma sullo Spectrum 16K dovrà separare queste tre sezioni, ed in particolare:

1) ISTRUZIONI

Battere linee 10, 60, 70, aggiungere la linea: 80 LOAD"" CODE:LOAD""
Battere linee da 2000 a 2850 e da 9500 alla fine.

Salvare quindi su nastro con SAVE "Paint 1" LINE 10

Verificare la correttezza della registrazione, quindi dare NEW

In questo modo si salvano le istruzioni e le routine atte a realizzare la simulazione del gioco.

2) UDG

Inserire come comando diretto CLEAR 32175

Battere le linee da 9000 a 9454.

Dare RUN e salvare con SAVE "Paint 2" CODE 32176,592 i caratteri immagazzinati.

Verificare registrazione, quindi NEW

In questo modo si salvano i caratteri grafici caricati dopo l'avvio del programma, non la routine di caricamento!

3) PROGRAMMA PRINCIPALE

Battere la linea: 15 LET z=0 : LET u=1 : LET c=5

Battere tutte le linee restanti, che non avete inserito nei 2 precedenti casi.

Salvare con SAVE "Paint 3" LINE 15

ATTENZIONE: nell'inserire il programma principale, per consentirne il funzionamento sullo Spectrum 16K, si rende necessario seguire alcuni accorgimenti: innanzitutto vanno rimossi tutti gli statement REM; in secondo luogo è opportuno raggruppare, dove è possibile, più linee di programma in una sola linea. Questa fase è molto delicata, nel senso che alcune istruzioni devono necessariamente essere in linee diverse. È ad esempio noto a tutti che gli statement IF... THEN non possono (meglio, non

dovrebbero!) essere riuniti in una sola linea di programma, poiché il verificarsi di combinazioni differenti di condizioni vere/false può produrre risultati diversi da quanto desiderato. Oltre a questo aspetto di carattere generale, ve n'è uno che riguarda specificamente Spepaint: alcune istruzioni devono risiedere necessariamente su linee di programma isolate. È il caso, ad esempio, delle cinque linee DATA 7149 - 7153, che non possono essere raggruppate, pur contenendo ciascuna un solo dato! Riunite dunque le istruzioni che possono essere palesemente raggruppate insieme (ad esempio, una routine per il sonoro che si esaurisce in poche linee di programma) e lasciate le cose come si trovano nei casi dubbi. Infine, sfruttando la linea 15 potrete risparmiare una quantità enorme di memoria. Per raggiungere tale obiettivo dovete sostituire nel listato tutti i valori numerici 0, 1 e 5 con le tre variabili ad essi associate. Ad esempio PLOT 5,0 deve diventare PLOT c,z. Il risparmio di memoria è dovuto al differente modo di immagazzinare variabili numeriche e valori costanti. Nell'esempio qui riportato il primo PLOT "costa" 17 byte, mentre il secondo ne richiede solo 5! Un'altra tecnica spesso sfruttata a tale scopo richiede l'uso di VAL; tale procedimento è più flessibile di quello prescelto per Spepaint, ma è meno "risparmioso".

Ancora una cosa

Concludiamo con un consiglio: vista l'indicizzazione di cui sopra, non è opportuno arrestare il programma con "BREAK" e poi dare nuovamente RUN. Infatti, a seconda del valore di determinati parametri, al momento dell'interruzione si potrebbero manifestare eventi indesiderati. Chi volesse comunque interrompere l'esecuzione del programma per dare un'occhiata ad un dato punto del listato abbia l'accortezza di eliminare almeno la linea 10 prima di lanciare nuovamente il programma con RUN. Meglio ancora, dopo un BREAK per ritornare a giocare inserite come comando diretto GO TO 7500: in tal modo attiverete la routine che vi richiede se volete giocare ancora. Scegliete l'opzione "Nuova Partita" e il programma procederà senza inconvenienti.

LOAD

Specpaint

Specpaint

Listato per ZX Spectrum 48K

```
1 REM *****
2 REM *< << SPEC-PAINT >> *>
3 REM *****
10 CLEAR USR "a"-425
20 LET ramtop=PEEK (23731)+2
30 LET rullo=0
40 LET costante=32382+(32768 AND ramtop>127)
45 POKE 23658,0: POKE 23609,10
50 DIM H$(5,6): LET max=0
60 DIM x$(5,10): DIM y$(5)
70 GO SUB 9500: REM ISTRUZIONI
80 LET g$="000000"
90 FOR f=1 TO 5: LET H$(f)=g$:
  LET X$(f)="-": LET Y$(f)="-": NEXT f
100 REM start
150 GO SUB 8000: REM DISPLAY
200 REM inizializzazione
210 LET l$="ABCDEFGHI"
220 LET b$="J{2G8}J{2G8}KII"
230 LET u$="LBCMEFNO{G8}"
240 LET p$=u$
300 LET ma=1: LET mb=16: LET ms=2:
  LET md=20
310 LET uscita=7000
320 LET sfondo=1
330 LET v=44: LET w=174
340 LET salto=0: LET raggio=1
350 LET qui=8: LET la=21
390 POKE 23675,176: POKE 23676,ramtop-2+(rullo=0)
400 REM *****
401 REM < LOOP PRINCIPALE >
402 REM *****
403 REM
500 IF RND<diffi THEN GO SUB (5000+salto)
600 PRINT AT qui,la; BRIGHT 1;
  PAPER sfondo; INK 0;p$(1 TO 3);
  AT qui+1,la;p$(4 TO 6);AT qui+2,
  la;p$(7 TO 9)
620 GO SUB uscita
650 IF INKEY$="" THEN GO TO 500
690 PRINT AT qui,la; PAPER sfondo;
  INK 0; BRIGHT 1;b$(1 TO 3);AT qui+1,la;b$(4 TO 6);AT qui+2,la;b$(7 TO 9)
700 LET flip=qui+3*(PEEK 23560=122
  AND qui<mb)-3*(PEEK 23560=97
  AND qui>ma)
710 IF flip<>qui AND ATTR (flip,la)=80 AND sfondo<>7 THEN LET flip=qui
```

```
720 LET flop=la+3*(PEEK 23560=107
  AND la<md)-3*(PEEK 23560=106
  AND la>ms)
730 IF flop<>la AND ATTR (flip,flop)=80 AND sfondo<>7 THEN
  LET flop=la
750 LET qui=flip: LET la=flop
800 IF la<20 THEN GO TO 1000
810 IF (qui=1 OR qui=4 OR qui=13
  OR qui=16) THEN LET la=la-3
820 IF (qui=7 OR qui=10) THEN LET qui=8: LET la=21: LET sfondo=1:
  LET p$=u$: LET uscita=7000:
  GO SUB 9900
1000 GO TO 400
2000 REM *****
2001 REM < DEMO ISTRUZIONI >
2002 REM *****
2003 REM
2010 BORDER 0: PAPER 0: INK 7: CLS
2020 PLOT 226,135: DRAW 0,-17: DRAW -52,0: DRAW 0,52: DRAW 52,0:
  DRAW 0,-17: INK 0
2030 INK 5: PLOT 226,140: DRAW 8,0:
  DRAW 0,-4: DRAW 8,8: DRAW -8,8:
  DRAW 0,-4: DRAW -8,0: INK 0
2040 FOR i=22582 TO 22742 STEP 32
2050 FOR f=i TO i+5
2060 LET pap=INT (RND*4+3)
2070 POKE f,pap*8+64
2080 NEXT f
2090 NEXT i
2100 RESTORE 2150
2110 FOR f=1 TO 5: READ o$
2120 PRINT-AT f*2-2,31; INK f+2;
  PAPER 0;o$
2130 NEXT f
2150 DATA "M","V","C","G","B"
2200 RETURN
2600 FOR f=22 TO 27
2610 LET inc=INT (RND*4)+3
2620 IF pag=2 THEN PRINT PAPER 0;
  INK inc;AT 8,f;"{G3}": GO TO 2640
2630 PRINT PAPER 0; INK inc;AT f-21,
  20;"{G5}"
2640 LET H$(1,f-21)=STR$ inc
2650 NEXT f
2700 RETURN
2800 FOR f=po TO ke STEP step
2810 POKE f,8*demo+64
2820 NEXT f
2850 RETURN
3000 REM *****
3001 REM < PUNTEGGIO PIU' ALTO >
3002 REM *****
3003 REM
3010 FOR f=1 TO 5
3020 LET max=VAL H$(f)
3030 IF punti>max THEN GO TO 3100
3050 NEXT f
3100 IF f=5 THEN GO TO 3200
3110 FOR t=5 TO f+1 STEP -1
3120 LET H$(t)=H$(t-1): LET X$(t)=X$(t-1): LET Y$(t)=Y$(t-1)
3130 NEXT t
```

```

3200 LET H$(f)=q$(1 TO 6-LEN q$)+q$
3210 INPUT "Il tuo nome, prego...";
      LINE o$
3220 LET X$(f)=o$: LET Y$(f)=STR$ liv
      ello
3300 BORDER 0: PAPER 0: INK 2: CLS
3310 PRINT AT 1,5;"I PUNTEGGI PIU' AL
      TI"
3320 PRINT AT 4,0; INK 7;"PUNTI";AT 4
      ,10;"GIOCATORE";AT 4,25;"LIVELLO
      "
3330 PRINT AT 5,0;"
      "
3340 INK 6
3350 FOR f=1 TO 5
3360 PRINT AT f*2+5,0;H$(f);AT f*2+5,
      10;X$(f);AT f*2+5,28;Y$(f)
3370 NEXT f
3380 FOR f=1 TO 300: NEXT f
3500 RETURN
4000 REM *****
4001 REM <      VITTORIA      >
4002 REM *****
4003 REM
4005 RESTORE 4100
4010 FOR f=1 TO 7
4020 READ durata,nota
4030 BEEP durata,nota
4040 NEXT f
4050 PAUSE 10
4060 BEEP .5,-7
4100 DATA .1,2,.075,0,.075,2,.075,0,.
      075,2,.075,0,.1,5
4110 FOR f=1 TO 30: OUT 254,RND*5+1:
      NEXT f
4150 BORDER 0
4160 FOR f=1 TO 100: NEXT f
4170 IF rullo=0 THEN LET rullo=1:
      GO TO 100
4200 LET imp=(65536*PEEK 23674+256*
      PEEK 23673+PEEK 23672)/50
4220 LET secondi=INT (imp-tempo)
4230 LET punti=INT (100000*coeff/seco
      ndi)
4240 POKE 23675,88: POKE 23676,ramtop
4250 BRIGHT 0: BORDER 1: PAPER 5:
      INK 0: CLS
4260 PLOT 206,72: DRAW 32,0: DRAW 0,3
      2,PI/4: DRAW -32,0: DRAW 0,-32,
      PI/4: DRAW 0,32,PI/4: PLOT 214
      ,88: DRAW -2,0: DRAW 0,-2:
      DRAW 2,0: DRAW 0,2: DRAW 8,16,
      PI/6.
4270 FOR f=5 TO 9 STEP 1: PLOT 206,72
      : DRAW 0,32,PI/f: DRAW 0,-32,
      PI/f: NEXT f: DRAW 0,32
4280 PLOT 40,8: DRAW -32,0: DRAW 0,32
      ,-PI/4: DRAW 32,0: DRAW 0,-32,
      PI/4: DRAW 0,32,PI/4: PLOT 30,
      24: DRAW -2,0: DRAW 0,-2: DRAW 2
      ,0: DRAW 0,2: DRAW -10,16,-PI/6
4290 FOR f=5 TO 9 STEP 1: PLOT 40,8:
      DRAW 0,32,PI/f: DRAW 0,-32,
      PI/f: NEXT f: DRAW 0,32
4300 REM Omini

```

```

4305 LET q$=STR$ punti: LET w$=STR$ s
      econdi
4310 LET a$="Secondi: "+w$+"Q"
4315 LET z$="Punti: "+q$
4320 LET p$=" L M": LET u$=" N O":
      LET par=1: LET fl=1: LET f2=23:
      LET f3=1: GO SUB 4350
4330 LET p$="L M ": LET u$="N O ":
      LET par=0: LET fl=23: LET f2=25
      -LEN a$: LET f3=-1: GO SUB 4350
4340 LET fl=24-LEN a$: LET f2=3:
      LET a$=a$( TO LEN a$-1)+
      "": GO SUB 4350
4345 GO TO 4400
4350 FOR f=fl TO f2 STEP f3
4360 PRINT AT 11,f;p$( TO 2);AT 12,f;
      p$(3 TO ): PAUSE 5
4370 PRINT AT 11,f;u$( TO 2);AT 12,f;
      u$(3 TO ): PAUSE 5
4375 IF par=1 THEN GO TO 4385
4380 PRINT AT 11,f+1;a$( TO 25-f-(1
      AND f2=3))
4385 NEXT f
4390 RETURN
4400 REM
4410 FOR f=LEN z$ TO 1 STEP -1
4420 PRINT AT 19,5;z$(f TO LEN z$):
      PAUSE 10
4430 NEXT f
4440 PRINT AT 19,5;"P"
4450 FOR f=6 TO 26-LEN z$
4460 PRINT AT 19,f+2;z$
4470 PRINT AT 19,f;" L";AT 20,f;" M":
      PAUSE 5
4480 PRINT AT 19,f;" N";AT 20,f;" O":
      PAUSE 5
4490 NEXT f
4500 REM verifica punteggio
4510 IF punti>=VAL H$(5) THEN
      GO SUB 3000
4600 REM nuova partita
4610 PRINT AT 21,0;"UN'ALTRA PARTITA
      ? (S/N)"
4620 INPUT LINE o$
4630 IF LEN o$>1 OR LEN o$<1 THEN
      GO TO 4620
4640 LET sicura=CODE o$(1)
4650 IF sicura<>110 AND sicura<>1
      15 THEN GO TO 4620
4660 IF sicura=115 THEN LET rullo=0:
      GO TO 100
4900 STOP
5000 REM *****
5001 REM <      QUADRATI ROSSI      >
5002 REM *****
5003 REM coordinate
5010 LET cx=(INT ((RND*6)+1))*3-1
5020 LET cy=(INT ((RND*6)+1))*3-1
5030 IF ATTR (cx,cy)=120 THEN
      RETURN
5040 LET salto=500
5050 RETURN
5500 REM disegno
5510 PRINT AT cx-1,cy; PAPER 2;
      BRIGHT 1;b$(1 TO 3);AT cx,cy;b$
      (4 TO 6);AT cx+1,cy;b$(7 TO 9)

```




```

5520 LET salto=0
5550 RETURN
7000 REM *****
7001 REM <USCITA SULLA FRECCIA>
7002 REM *****
7003 REM
7005 LET cassetti=costante: LET verifica=0
7010 FOR y=2 TO 17 STEP 3
7020 FOR x=2 TO 17 STEP 3
7030 IF ATTR (x,y)=PEEK (cassetti)
      THEN LET verifica=verifica+1
7040 LET cassetti=cassetti+(rullo=0)
7045 IF INKEY$="0" THEN GO TO 7500
7050 IF INKEY$<>" " THEN GO TO 7090

7060 NEXT x
7065 LET cassetti=cassetti+(rullo=1)-6*(rullo=0)
7070 NEXT y
7080 IF verifica=36 THEN GO TO 4000
7090 IF PEEK 23560>=49 AND PEEK 23560<=53 THEN LET sfondo=(PEEK 23560)-46: LET uscita=7777:
      RESTORE 7100+PEEK 23560: READ 1
      : GO SUB 7200: GO SUB 7300:
      LET p$=1$: LET qui=7+INT (RND*2)
      )*3: LET la=17: RETURN
7100 GO TO 7000
7149 DATA 172
7150 DATA 140
7151 DATA 108
7152 DATA 76
7153 DATA 44
7200 REM selezione vernice
7210 OVER 1: PLOT w,v: GO SUB 7250:
      OVER 0
7230 INK 7
7235 LET v=1: LET w=221
7240 PLOT w,1
7250 DRAW 28,0: DRAW 0,-32: DRAW -28,0: DRAW 0,32
7260 INK 0: RETURN
7300 REM rientro
7310 FOR f=8 TO 10
7320 PRINT AT f,21; BRIGHT 1; PAPER 1;
      " "
7330 NEXT f
7340 RETURN
7500 REM fine gioco?
7510 POKE 23560,0
7520 LET o$="PREMI <S> PER UN'ALTRA PARTITA PREMI <N> SE VUOI TERMINARE PREMI <C> SE VUOI CONTINUARE "
7530 FOR f=1 TO 1000
7535 LET inc=4
7540 FOR i=1 TO 65 STEP 32
7550 PRINT AT 21,0; INK inc;o$(i TO i+31): FOR j=1 TO 70: NEXT j:
      BEEP .01,4*inc: LET inc=inc+1
7560 GO TO 7580+50*(PEEK 23560=115)+30*(PEEK 23560=110)+40*(PEEK 23560=99)
7580 NEXT i
7600 NEXT f

```



```

7610 STOP
7620 PRINT AT 21,0;"
      ": GO TO 7000
7630 LET rullo=0: GO TO 100
7777 RETURN
8000 REM *****
8001 REM <ALLESTIMENTO SCHERMO>
8002 REM *****
8003 REM
8004 POKE 23675,88: POKE 23676,ramtop

8005 BORDER 0: PAPER 0: BRIGHT 1:
      CLS
8010 INK 7: PLOT 163,115: DRAW 0,56:
      DRAW -150,0: DRAW 0,-150:
      DRAW 150,0: DRAW 0,56
8020 INK 0: FOR x=16 TO 136 STEP 24:
      PLOT x,24: DRAW 0,142: NEXT x
8030 FOR y=24 TO 144 STEP 24: PLOT 16,y:
      DRAW 142,0: NEXT y
8100 LET a$="{3G8}"
8110 FOR x=1 TO 16 STEP 3
8120 FOR y=2 TO 17 STEP 3
8130 LET pap=INT (RND*4)+3
8140 PAPER pap
8150 LET as=x: LET or=y
8160 FOR j=1 TO 3
8170 OVER 1
8180 PRINT AT as,or;a$
8190 LET as=as+1
8200 NEXT j
8205 GO SUB 9900
8210 NEXT y
8220 NEXT x
8400 REM freccia
8410 INK 0: OVER 0
8420 PAPER 1: BRIGHT 1
8430 FOR f=8 TO 11
8440 PRINT AT f,20;"{4G8}"
8450 NEXT f
8460 PRINT AT 6,21;"{G8}";AT 7,21;"
      {2G8}";AT 9,24;"{G8}";AT 10,24;"
      {G8}";AT 12,21;"{2G8}";AT 13,21;"
      "{G8}"
8470 FOR f=5 TO 9: PRINT AT f,f+16;"J
      ";AT 19-f,f+16;"K"
8480 NEXT f
8490 BRIGHT 0
8500 REM barattoli
8510 PAPER 0
8520 LET inc=2
8530 FOR f=1 TO 17 STEP 4
8540 LET inc=inc+1
8550 PRINT AT f,28; INK inc;"ABC";
      AT f+1,28;"DEF";AT f+2,28;"GHI";
      AT f+3,28;" "
8560 NEXT f

```

```

8600 REM indici per il colore
8610 LET cassette=costante
8620 FOR f=2 TO 17 STEP 3
8630 LET pap=INT (RND*4)+3
8640 IF rullo=0 THEN PRINT PAPER pa
p;AT f-1,0;"{G8}";AT f,0;"S";
AT f+1,0;"{G8}": GO TO 8660
8650 PRINT AT 20,f; PAPER pap;"{G8}↑
{G8}"
8660 POKE cassette,8*pap+64
8670 LET cassette=cassette+1
8680 NEXT f
8700 REM livelli di gioco
8710 IF rullo=1 THEN RETURN
8720 INK 7: INPUT "LIVELLO DI DIFFICO
LTA' (1/4)? "; LINE o$
8730 IF LEN o$>1 OR LEN o$<1 THEN
GO TO 8720
8740 LET sicura=CODE o$(1)
8750 IF sicura<49 OR sicura>52 THEN
GO TO 8720
8760 LET livello=VAL o$
8770 LET diffi=livello*.025
8780 LET coeff=livello*2
8790 LET tempo=(65536*PEEK 23674+256*
PEEK 23673+PEEK 23672)/50
8800 REM si torna alla linea 200
ed inizia il gioco
8810 INK 0
8820 RETURN
9000 REM *****
9001 REM < U.D.G. >
9002 REM *****
9003 REM
9005 POKE 23675,88: POKE 23676,ramtop

9010 RESTORE 9200
9020 FOR f=USR "a" TO USR "s"+7
9030 READ udg1: POKE f,udg1
9040 NEXT f
9045 RETURN
9050 POKE 23675,176: POKE 23676,ramto
p-1
9060 RESTORE 9300
9070 FOR f=USR "a" TO USR "o"+7
9080 READ udg2: POKE f,udg2
9090 NEXT f
9095 RETURN
9100 POKE 23675,176: POKE 23676,ramto
p-2
9110 RESTORE 9400
9120 FOR f=USR "a" TO USR "o"+7
9130 READ udg3: POKE f,udg3
9140 NEXT f
9150 RETURN
9200 REM 1' Tavola UDG
9210 REM barattoli (9 data)
9211 DATA 0,0,0,63,63,47,32,40
9212 DATA 0,0,255,255,255,255,126,0
9213 DATA 0,0,0,252,252,244,4,12
9214 DATA 36,35,32,32,32,32,32,32
9215 DATA 0,128,127,0,0,0,0,0
9216 DATA 20,36,176,4,4,4,4,4
9217 DATA 32,32,32,32,30,1,0,0
9218 DATA 0,0,0,0,129,126,0,0
9219 DATA 4,4,4,4,120,128,0,0

```

```

9230 REM freccia (2 data)
9231 DATA 255,127,63,31,15,7,3,1
9232 DATA 1,3,7,15,31,63,127,255
9250 REM omino (4 data)
9251 DATA 24,60,60,16,56,52,54,48
9252 DATA 48,56,40,36,36,66,130,130
9253 DATA 6,15,15,4,14,15,14,30
9254 DATA 12,12,12,10,10,10,18,18
9270 REM varie (p,q,r,s)
9271 DATA 224,224,224,224,224,208,208
,208
9272 DATA 31,31,31,31,31,31,15,15
9275 DATA 8,30,61,126,124,60,104,192
9280 DATA 0,0,8,4,126,4,8,0
9300 REM 2' Tavola UDG
9310 REM pennello 1 (9 data)
9311 DATA 128,128,128,128,128,128,129
,132
9312 DATA 0,0,8,28,46,91,148,231
9313 DATA 0,0,0,0,0,0,128,160
9314 DATA 135,137,144,144,136,132,130
,131
9315 DATA 254,163,245,119,63,120,113,
7
9316 DATA 224,48,184,208,160,192,128,
0
9317 DATA 143,144,161,193,194,172,144
,255
9318 DATA 129,196,56,0,0,0,0,255
9319 DATA 0,0,0,0,0,0,0,255
9330 REM bordini (2 data)
9331 DATA 128,128,128,128,128,128,128
,128
9332 DATA 128,128,128,128,128,128,128
,255
9350 REM pennello 2 (4 data)
9351 DATA 0,0,0,0,0,0,1,3
9352 DATA 7,9,16,16,8,4,2,3
9353 DATA 15,16,33,65,66,44,16,0
9354 DATA 129,196,56,0,0,0,0,0
9400 REM 3' Tavola UDG
9410 REM rullo 1 (9 data)
9411 DATA 128,129,158,174,196,200,144
,144
9412 DATA 192,96,56,12,3,1,0,0
9413 DATA 0,0,0,0,0,128,192,112
9414 DATA 136,230,179,140,134,131,130
,134
9415 DATA 0,0,0,128,64,32,152,68
9416 DATA 48,12,4,8,16,56,88,128
9417 DATA 136,144,161,193,194,236,240
,255
9418 DATA 131,128,0,0,0,0,0,255
9419 DATA 0,0,0,0,0,0,0,255
9430 REM bordini (2 data)
9431 DATA 128,128,128,128,128,128,128
,128
9432 DATA 128,128,128,128,128,128,128
,255
9450 REM rullo 2 (4 data)
9451 DATA 0,1,30,46,68,72,144,144
9452 DATA 136,102,51,12,6,3,2,6
9453 DATA 8,16,33,65,66,108,112,0
9454 DATA 131,128,0,0,0,0,0,0
9500 REM *****
9501 REM < ISTRUZIONI >

```

LOAD

Specpaint

```
9502 REM *****
9503 REM
9510 GO SUB 2000
9520 INK 7
9530 PRINT AT 0,4;"SPEC-PAINT"
9540 LET pag=1
9550 PRINT AT 3,0;"Il gioco e' diviso
      ""in due fasi :""-Nel corso
      della" PAPER 1; FLASH 1;
      BRIGHT 1;"PRIMA FASE"; PAPER 0;
      FLASH 0; BRIGHT 0;" dovrai"
9560 PRINT "spostare lungo la""scacc
      hiera di gioco""un grosso penne
      llo, con il qualetenterai di ve
      rnicciare le sei linee orizzonta
      li secondo quantoriportano gli i
      ndici di colore asinistra della
      scacchiera."
9570 PAUSE 200: PRINT ""Ora ti mostr
      o:" PAUSE 50: PRINT "Guarda la
      scacchiera, in alto a destra sul
      lo schermo": PAUSE 100
9575 LET g=7
9580 GO SUB 2600: GO SUB 9900
9590 FOR t=22747 TO 22587 STEP -32
9595 LET step=-1: LET po=t: LET ke=po
      -5: POKE (po-7),PEEK (po-7)+128
9600 LET g=g-1: LET demo=VAL H$(1,g)
9605 GO SUB 2800: GO SUB 9900:
      PAUSE 30
9610 NEXT t
9615 GO SUB 9000: FOR f=22580 TO 2274
      0 STEP 32: POKE f,PEEK (f)-128:
      NEXT f: GO SUB 9950
9620 LET pag=2
9630 INK 7: PRINT AT 3,0;"-Nel corso
      della" PAPER 1; FLASH 1;
      BRIGHT 1;"SECONDA FASE";
      PAPER 0; FLASH 0; BRIGHT 0;" do
      vrai""invece verniciare le""se
      i colonne verticali""muovendo u
      n grosso""rullo da imbianchino.
      "
9640 PAUSE 50: PRINT ""Anche questa
      volta ti porterò un piccolo es
      empio:"; PAUSE 50: PRINT " tu g
      uarda""la scacchiera!": PAUSE 5
      0
9650 LET g=7: GO SUB 2600: GO SUB 990
      0
9655 FOR t=22747 TO 22742 STEP -1
9660 LET step=-32: LET po=t: LET ke=p
      o-160: POKE (po+64),PEEK (po+64)
      +128
9665 LET g=g-1: LET demo=VAL H$(1,g)
9670 GO SUB 2800: GO SUB 9900:
      PAUSE 30
```

```
9675 NEXT t
9680 GO SUB 9050: FOR f=22806 TO 2281
      1: POKE f,PEEK (f)-128: NEXT f:
      GO SUB 9950
9690 INK 7: PRINT AT 3,0;"Potrai sele
      zionare""il colore della""v
      ernice portandoti""sulla frecci
      a blu,""a destra della""sc
      acchiera, quindi""premendo un t
      asto fra 1 & 5, abbinati risp
      ettivamente alle vernici ";
      PAPER 3; INK 0;"MAGENTA";
      PAPER 0; INK 7;","; PAPER 4;
      INK 0;"VERDE"; PAPER 0; INK 7;"
      ,"; PAPER 5; INK 0;"AZZURRA";
      PAPER 0; INK 7;","
9695 PRINT INK 0; PAPER 6;"GIALLA";
      PAPER 0; INK 7;" & "; INK 0;
      PAPER 7;"BIANCA"; PAPER 0;
      INK 7;". In particolare l'ultim
      a vernice serve soltanto a cance
      llare macchie "; PAPER 2;"ROSSE"
      ; PAPER 0;" che""appaiono qua
      e la ', e che costituiscono
      l'unico ostacolo al completamen
      to della partita:"
9700 PRINT "infatti esse non possono
      essere cancellate che con una en
      ergica mano di bianco!"
9710 GO SUB 9100: GO SUB 9950
9720 INK 4: PRINT AT 11,0;"Al termine
      di ogni fase riporta il pennell
      o o il rullo sopra la freccia bl
      u e attendi : verra' verificata
      la correttezza della tua vernic
      iatura in modo che tu possa acc
      edere alla fase successiva
      o al quadro ove ti saranno co
      municati il punteggio realizzato
      e il tempo impiegato."
9730 GO SUB 9950
9740 POKE 23675,88: POKE 23676,ramtop
9750 INK 7: PAPER 1: FLASH 1: PRINT
      AT 3,0;"*TASTI>"; INVERSE 1;"<MO
      VIMENTO*"
9755 FLASH 0: RESTORE 9765
9760 FOR f=1 TO 4: READ d1,d2: PLOT d
      1,d2: DRAW 0,24: DRAW 24,0:
      DRAW 0,-24: DRAW -24,0: NEXT f
9765 DATA 24,48,40,16,152,48,184,48
9770 PRINT AT 14,4;"A";AT 18,6;"Z";
      AT 14,20;"J";AT 14,24;"K"
9775 LET qui=11: LET la=14
9780 PRINT AT qui+5,la;"R"
9790 RESTORE 9830: FOR f=5 TO 11
      STEP 2: READ s$: READ s1,o1,o2,
      t1,t2,t3
9795 INK 7: PRINT AT f,0;s$(1 TO 7);:
      FOR k=0 TO 64 STEP 32: FOR j=0
      TO 2: POKE s1+j+k,120: NEXT j:
      NEXT k: GO SUB 9900
9800 PRINT INK 2;s$(8 TO LEN s$)
9805 IF o2=1 THEN LET qui=qui+5
9810 GO SUB 9930
```

Novità firmate Jackson.

Le Novità Jackson puoi acquistarle in via Mascheroni 14 a Milano e in tutte le migliori librerie.



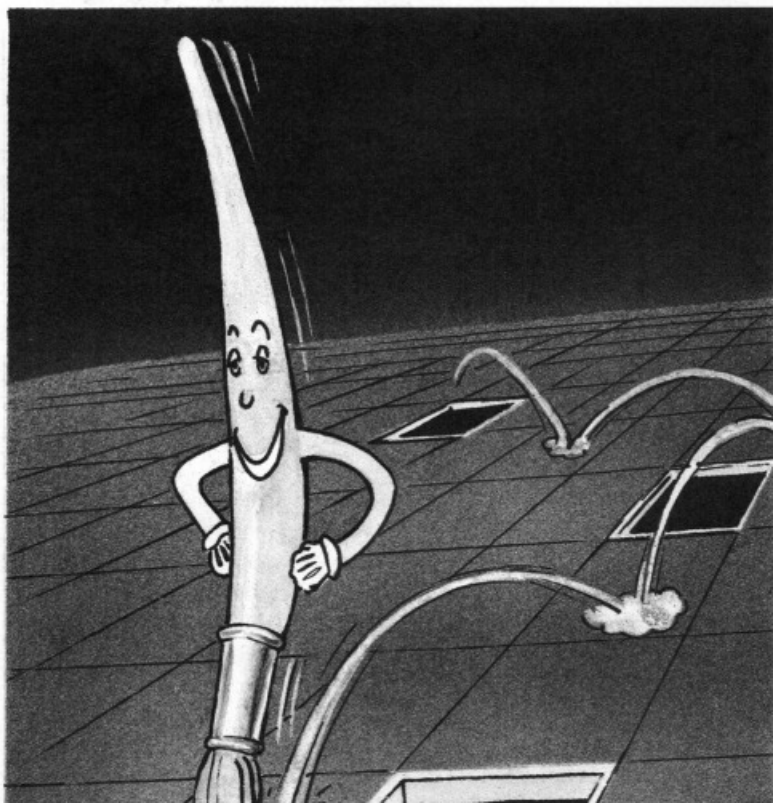
GRUPPO
EDITORIALE
JACKSON

La biblioteca che fa testo.

```
9815 IF t>=18 THEN PRINT AT 16,31;"
      ": GO TO 9870
9820 FOR i=1 TO 200: NEXT i
9825 GO SUB 9900: FOR k=0 TO 64
      STEP 32: FOR j=0 TO 2: POKE s1+
      j+k,56: NEXT j: NEXT k
9830 DATA "A.....ALTO",22947,1,0,5,1
      ,-1,"Z.....BASSO",23077,-1,0,1,
      5,1,"J.....SINISTRA",22963,0,1,
      0,-5,-1,"K.....DESTRA",22967,0,
      -1,-5,17,1
9835 NEXT f
9840 RETURN
9900 REM sonoro
9910 BEEP .002,30: BEEP .01,20:
      BEEP .002,40: BEEP .003,50
9920 RETURN
9930 REM demo movimento
9940 FOR t=t1 TO t2 STEP t3: PRINT
      AT qui+(t AND o2=0),la+(t AND o1
      =0);"R";AT qui+o1+(t AND o2=0),l
      a+o2+(t AND o1=0);" ": NEXT t
9945 RETURN
9950 REM cls pagine istruzioni
9960 FOR i=1 TO 1000
9965 FOR f=1 TO 6
9970 IF INKEY$<>" " THEN GO TO 9985

9975 INK f: PRINT AT 21,0;"PREMI UN
      TASTO PER CONTINUARE": PAUSE 5

9980 NEXT f: NEXT i
9985 LET r$="
      "
9990 FOR f=1 TO 8: PRINT AT f,0;r$(
      TO 21): NEXT f
9995 FOR f=9 TO 21: PRINT AT f,0;r$:
      NEXT f
9999 RETURN
```

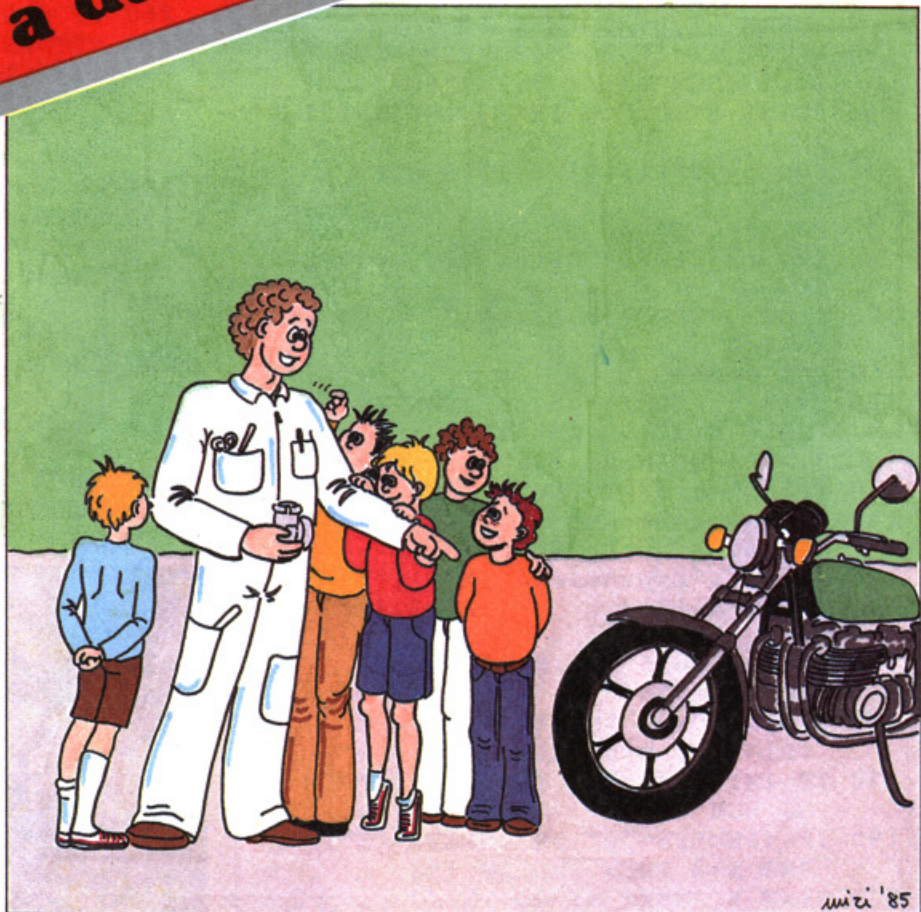


Motore a due tempi

di **Tim Rose**
trad. e adatt.
di **Angela Spinazzola**

Quanti di voi hanno visto l'interno di un motore in funzionamento?

Sulle strade di tutto il mondo sono milioni i motorini che scorrazzano guidati da giovani centauri e molti di questi sono anche bravi a smontarli e rimontarli nel tentativo di spremere qualche stilla di velocità in più. Libri sui motori ve ne sono molti, ampiamente illustrati, ma nessuno di essi può mostrarvi il motore in movimento. L'aiuto di disegni in sequenza è efficace, ma... con lo Spectrum è un'altra cosa. Ecco il vostro generoso motore che si muove, visualizza le fasi di funzionamento e si arresta docile, se volete riflettere su qualche particolare. "S" è il tasto di arresto. Gli studenti di scuola media non dovrebbero mancare di portarsi in aula il programma quando l'insegnante di educazione tecnica tratterà l'argomento.



Descrizione del programma

Linee
20-60
70-200
210-220
230-270
290-300

Commento
disegno del motore
ciclo di movimento
aggiornamento immagine. Fase di scoppio/compressione
aggiornamento immagine. Fase di scarico/aspirazione
caratteri grafici

Motore a 2 tempi versione per ZX Spectrum 16K/48K

```
5 CLS : REM motore a 2 tempi
6 PRINT "CON QUESTO PROGRAMMA HAI LA POS-SIBILITA' DI VEDERE IL FUNZIONAMENTO DEL MOTORE A SCOPPIO CON CICLO A 2 TEMPI""SE TI INTERESSA OSSERVARE CON ATTENZIONE UNA FASE PUOI FERMARE IL PROGRAMMA PREMENDO ~S~.""RILASCIANDO IL TASTO IL PROGRAMMA RIPRENDE A GIRARE""premi un tas
```

```
to per incominciare.": PAUSE 0
10 GO SUB 290: BORDER 0: PAPER 0:
   INK 6: CLS
20 CIRCLE 125,39,24
30 PRINT AT 20,0;"Motore"" a 2 tempi"
40 PLOT 102,160: DRAW 0,-50: DRAW -12,-15: DRAW 0,-50: DRAW 70,0,
   PI: DRAW 0,30: DRAW -8,5:
   DRAW 15,0: PLOT PEEK 23677,
   PEEK 23678+12
50 DRAW -15,0: DRAW 0,4: DRAW 15,0:
   PLOT PEEK 23677,PEEK 23678+12:
```

LOAD

**Motore
a due tempi**

```

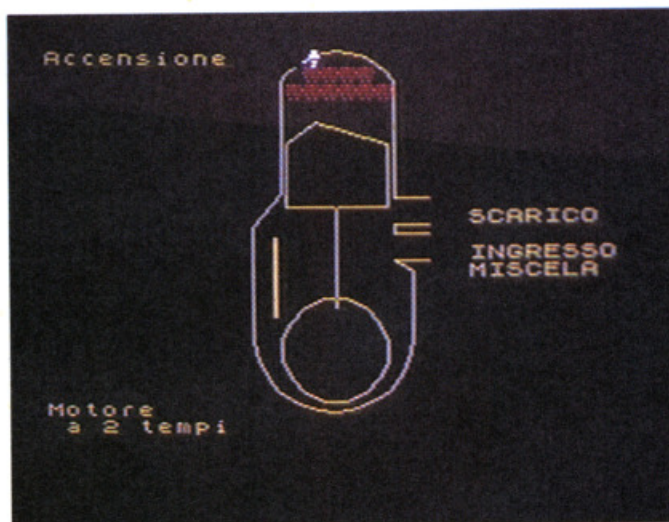
DRAW -15,0: DRAW 0,53: DRAW -50
,0,2: PLOT 100,90: DRAW 0,-35:
DRAW -1,0: DRAW 0,35
55 PRINT AT 11,23;"INGRESSO";AT 12,
23;"MISCELA";AT 9,23;"SCARICO"
60 PRINT AT 0,14; INK 7; BRIGHT 1;"
A"
70 FOR a=(2*PI)-PI/6 TO 0 STEP -
PI/6
80 FOR z=0 TO 1
90 LET b=126+20*COS a: LET c=39+20*
SIN a
100 PLOT OVER 1;b,c: DRAW OVER 1;1
26-b,45
110 LET d=c+45
120 OVER 1: PLOT 105,d: DRAW 44,0:
DRAW 0,28: DRAW -32,10: DRAW -1
3,-10: DRAW 0,-28: OVER 0
130 IF INT c=59 THEN GO SUB 210
140 IF INT c=56 AND b>126 THEN
PRINT AT 0,0;"Scoppio ";
AT 1,0;"Espansione";AT 1,14;"
";AT 2,13;" ";AT 3,13;"
"
150 IF INT c=21 AND b>126 THEN
GO SUB 230
160 IF INT c=38 AND b<126 THEN
PRINT AT 0,0;"Compressione"
170 IF INT c=48 AND INT b<126 THEN
GO SUB 280
190 IF (INKEY$="s" OR INKEY$="S")
AND z=0 THEN GO TO 190
200 NEXT z: NEXT a: GO TO 70
210 PRINT AT 0,0;"Accensione ";
AT 1,14; INK 2; BRIGHT 1;"BBBBB";
AT 2,13;"BBBBBB": IF z=0 THEN
FOR x=0 TO 10: BEEP .01,10:
BEEP .01,20: NEXT x
220 RETURN

```

```

230 PRINT AT 0,0;" ""
": OVER 1: PLOT 134,122:
DRAW 20,-20,PI/2: PLOT 155,102
: DRAW 16,0: PLOT 166,106:
DRAW 5,-4: DRAW -5,-4
240 PRINT AT 9,0;"Aspirazione""forz
ata": PLOT 96,40: DRAW 0,55:
DRAW 20,22: DRAW -5,0: POKE 236
77,PEEK 23677+5: DRAW 0,-5:
OVER 0
260 IF z=0 THEN NEXT z
270 RETURN
280 OVER 1: PLOT 160,85: DRAW -12,0:
DRAW -5,-20: DRAW -4,5: POKE 23
677,PEEK 23677+11: DRAW -6,-5:
OVER 0
281 GO TO 250
290 RESTORE 300: FOR x=0 TO 14:
READ y: POKE USR "a"+x,y:
NEXT x: RETURN
300 DATA 14,14,31,31,32,206,4,12,90,
0,169,0,74,0,36,0
9000 CLS : PRINT FLASH 1;AT 10,0;"
FERMA IL NASTRO "
9010 PRINT AT 21,0;"Premi un tasto pe
r incominciare.": PAUSE 0: RUN
9990 SAVE "motore2T" LINE 9000
9995 VERIFY ""
9999 STOP

```



**Dalla grande
edicola
Jackson**

**Tutte le applicazioni
professionali**

AUTOMAZIONE

Un'aggiornatissima panoramica delle nuove tecnologie microelettroniche e informatiche applicate all'automazione industriale.
11 numeri all'anno: L. 3.500 a numero
Abbonamento: solo L. 30.500

electronica
OGGI

Il punto di riferimento più qualificato per chi voglia aggiornarsi su prodotti, applicazioni, tecnologie elettroniche, in Italia e all'estero.
11 numeri all'anno: L. 3.500 a numero
Abbonamento: solo L. 31.000

telecomunicazioni

Le frontiere aperte dalla telematica, le telecomunicazioni professionali in tutti i loro settori.
10 numeri all'anno: L. 3.500 a numero
Abbonamento: solo L. 28.000

INFORMATICA

La rivista professionale per chi si occupa di sistemi: dai microcomputer ai mini, ai supermini, ai mainframe. Con notizie in anteprima dall'America.
11 numeri all'anno: L. 3.500 a numero
Abbonamento: solo L. 31.000

l'Electronica

Quindicinale di politica industriale, componentistica, informatica e telecomunicazioni per uomini di marketing, responsabili acquisti, manager di settore.
22 numeri all'anno: L. 2.500 a numero
Abbonamento: solo L. 44.000

**Quando l'informazione
fa testo**

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 MI

Desidero ricevere GRATIS un numero
della Rivista _____

(allego L. 1.000 in francobolli per
contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della
Biblioteca JACKSON (allego L. 1.000
in francobolli per contributo spese di
spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

Slalom

di **Andrew Norton**
trad. e adatt.
di **Guido Bertolotti**

Un simpatico gioco pensato per lo ZX81 inespanso.

Questo programma gira su ZX81 anche senza espansione di memoria ed è sufficientemente veloce, pur funzionando in modo SLOW.

Lo scopo di questo gioco è di guidare uno sciatore, che scende da una collina, attraverso venti porte posizionate irregolarmente senza urtarne i pali. Lo sciatore si controlla con i tasti 5 e 8.

Alla fine viene mostrato il punteggio: se maggiore di 15, è piuttosto buono.

Per rendere il gioco più facile si può modificare la larghezza delle porte: per allargarle, ad esempio, di due caratteri basta aggiungere due "-" alla linea 50 e cambiare la linea 120 in: IF X=Y OR X=Y+1 OR X=Y+2 THEN LET S=S+1.

Per aumentare o diminuire il numero delle porte, che in origine sono 20, basta modificare la linea 140.

T è il numero di porte incontrate, S il punteggio, X la posizione (colonna) dello sciatore, Y la posizione (colonna) della porta.



Slalom versione per ZX81 1K

```

5  RAND
10  LET S=0
20  LET T=0
30  LET X=14
40  LET Y=INT (RND*26)
50  PRINT AT 21,Y;"*----*"
60  FOR N=1 TO 11
70  GOSUB 400
80  SCROLL
90  SCROLL
100 GOSUB 400
110 NEXT N
    
```

```

120 IF X=Y THEN LET S=S+1
130 LET T=T+1
140 IF T=20 THEN GOTO 300
150 GOTO 40
300 CLS
310 PRINT "PUNTEGGIO: ";S;" SU
";T
320 STOP
410 IF INKEY$="5" AND X>0 THEN
LET X=X-1
420 IF INKEY$="8" AND X<26 THEN
LET X=X+1
430 PRINT AT 0,X;" 0 ";TAB X;
";" ";TAB X;" )";TAB X;" )"
460 RETURN
    
```

LOAD

Saltamassi

di **Luuk Hilhorst**
trad. e adatt.
di **Guido Bertolotti**

Mettete alla prova il vostro colpo d'occhio: basta 1k...

In questo semplice, ma avvincente gioco dovete saltare i massi che vi rotolano addosso, per evitare di essere schiacciati.

Per saltare premete il tasto O.

Il gioco termina con la dolorosa collisione con un masso, dopodiché viene mostrato il punteggio.

Per ricominciare a giocare basta premere un tasto qualunque.

Descrizione del programma

Linee	Commento
1-20	inizializzazione delle variabili
40	cambia la posizione dei massi
60-70	controllano la chiamata della subroutine per il salto
80	visualizza i massi nella nuova posizione
85	incrementa il punteggio
90-96	controllo della collisione con un masso. La linea 90 sposta la posizione di PRINT alle coordinate X, 10: in pratica il suo effetto è di mettere nella variabile di sistema DF_CC (byte 16398-16399) l'indirizzo del byte in cui verrà stampato il successivo carattere. La linea 95 prende questo indirizzo e con esso va a leggere il contenuto del byte in cui dovrà essere stampato l'omino: se il contenuto è CODE "O" (cioè un masso) il programma passa alla linea 110; altrimenti (linea 96) viene stampato l'omino nella nuova posizione
100-104	subroutine del salto
110-140	collisione e fine gioco

Saltamassi versione per ZX81 1K

```

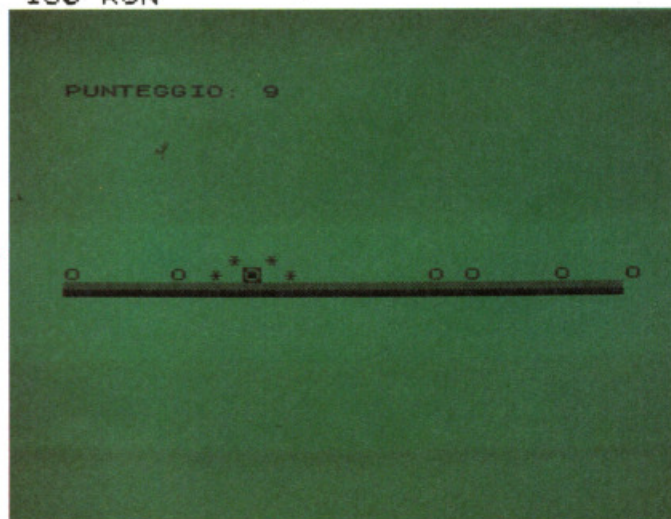
1 LET X=VAL "12"
2 LET S=NOT PI
3 FOR I=NOT PI TO CODE "2"
5 PRINT AT CODE "$",I;" "
6 NEXT I
10 LET A$=" 0 0 0 0 0
   0 0"
20 LET A$=A$+A$
30 LET A=NOT PI
40 LET A=A+1
45 PRINT AT X,10;" "
50 IF A=31 THEN GOTO VAL "30"
60 IF X=12 AND INKEY$="O" THEN
GOSUB 100
70 IF X=11 THEN GOSUB 101
80 PRINT AT 12,0;A$(A TO A+31)
85 LET S=S+1
90 PRINT AT X,10;
95 IF PEEK (256*PEEK 16399+PEE
K 16398)=CODE "O" THEN GOTO VAL
"110"
96 PRINT "I"
97 GOTO 40
100 LET Y=VAL "4"
101 LET Y=Y-SGN PI
102 IF Y=NOT PI THEN LET X=VAL
"12"

```

```

103 IF Y>NOT PI THEN LET X=11
104 RETURN
110 PRINT AT X-SGN PI,VAL "9";"
* *";TAB VAL "8";"* * "
120 PRINT AT NOT PI,NOT PI;"PUN
TEGGIO: ";S
130 PAUSE VAL "4E4"
140 CLS
150 RUN

```



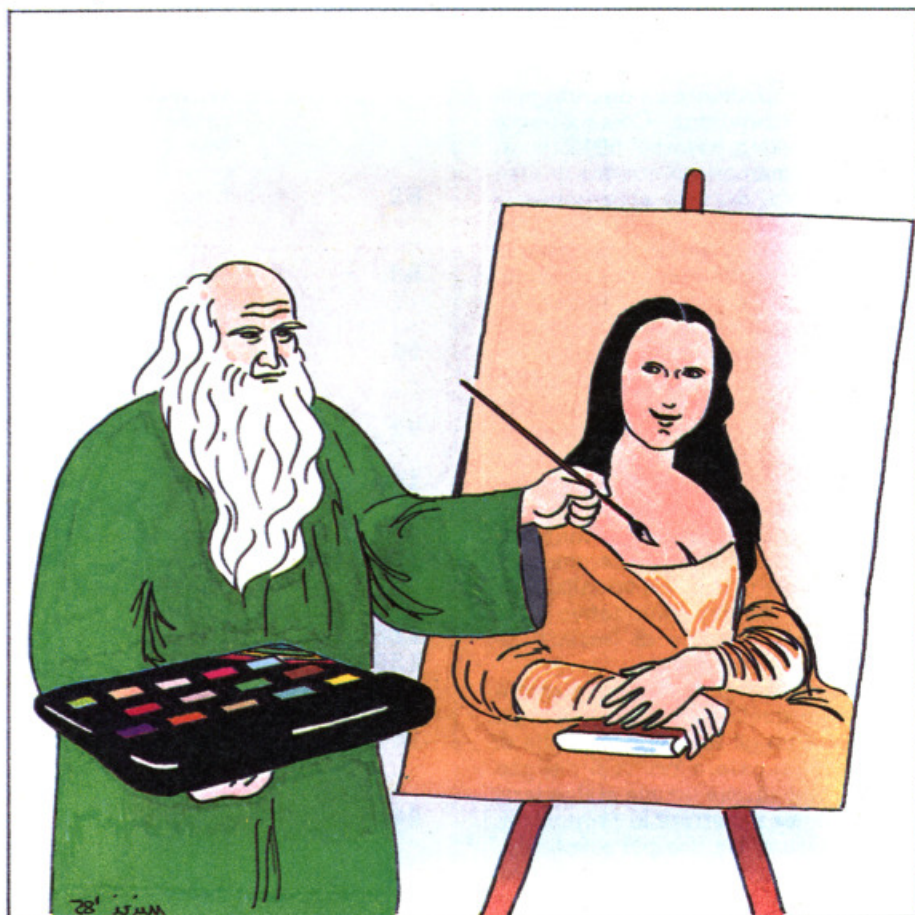
Tavolozza

di Dino Citterio

Realizzato con intenti divulgativi, ecco un programma per disegnare con lo Spectrum che interesserà tutti i programmatori BASIC.

Vi sono in commercio non pochi programmi per disegnare con lo Spectrum e sembrerebbe fatica inutile presentarne un altro, addirittura meno sofisticato e più lento in esecuzione. Eppure eccolo qui, sulle pagine di una rivista di pregio, offrirsi alla vostra attenzione per una somma di motivi che non è inutile accennare. Entrare in un negozio e portarsi a casa la cassetta di una software house non comporta oggi una spesa proibitiva. I risultati sono garantiti e dopo aver ingoiato le migliaia di byte che essa contiene il fedele Spectrum è pronto a offrirvi tutte le sue funzioni, sicuro e veloce così come il linguaggio macchina glielo ordina. La professionalità che sta dietro al programma è garanzia di buoni risultati, ma vi lascia tra le mani un prodotto freddo, del cui funzionamento difficilmente vi renderete conto anche se andrete a leggere tutti i recessi della memoria, stanando l'infinita serie di numeri che essa contiene. Non vogliamo prendere in considerazione l'eventualità di qualche "baco" nel programma, perché il linguaggio macchina che lo supporta renderebbe pressoché impossibile rimediare al problema.

Osservazione conclusiva, con la quale termino la maldicenza verso il linguaggio macchina d'acquisto, è che non saprete mai attraverso quali ragionamenti il programmatore è giunto a darvi il suo prodotto. A fronte di ciò il listato che avete fra le mani. È praticamente tutto in BASIC, essendo stato limitato il ricorso al linguaggio macchina solo ad una breve routine per il trasferimento in memoria e per il richiamo dell'immagine video. Il programma si rivolge a quanti



vogliono accostarsi alla programmazione, soprattutto ai più giovani e specialmente agli insegnanti della scuola dell'obbligo che vi troveranno infiniti spunti di utilizzo.

Funziona così...

Diciamo subito che la velocità di esecuzione non è fulminea. È stata una scelta di fondo iniziale, utile per consentire all'utente (e continuo a pensare allo studente) di capire ciò che succede premendo un certo tasto, specialmente se avrà l'accortezza di lavorare con a fianco il listato. D'altra parte viaggiamo con il BASIC, da sempre accusato, a torto o a ragione, di essere un linguaggio lento.

Le funzioni offerte sono molte, ed altre ne potevano essere aggiunte. Se non l'ho fatto è stato per non appesantire

ulteriormente il listato, visto e considerato che quelle fondamentali, a mio avviso, ci sono tutte. Forse potevano entrarci ancora due routine per disegnare un arco di cerchio e il rettangolo. Chi avrà la pazienza di caricare il listato, rendendosi conto della struttura e delle scelte di programmazione, non avrà difficoltà, se proprio non è digiuno di BASIC, a colmare la lacuna. E già che ci siamo offriamo qualche altra idea: disegnare triangolo, quadrato e cerchio in posizione diversa rispetto al punto di PLOT, prevedere anche il triangolo isoscele oltre a quello equilatero, inserire una routine per copiare nella metà inferiore dello schermo ciò che compare in quella superiore (attributi compresi), offrire la possibilità di definire caratteri grafici da usare nel disegno. Infine si potrebbe prevedere una funzione per tracciare la cornice al disegno invece di chiamare

quattro volte la funzione linea. Per i volenterosi mi pare che basti.

Entriamo ora nella struttura del programma. Lo possiamo considerare diviso in quattro blocchi fondamentali. La presentazione con le istruzioni (linee 1-80 e 1450-1485), il ciclo di lavoro per la lettura della tastiera e la realizzazione del disegno (linee 81-500), le subroutine per l'esecuzione delle varie funzioni (linee 1000-2380), i tre blocchi di memoria ove vengono ospitati ben tre schermi di lavoro, richiamabili a volontà. Un quarto blocco conserva lo schermo corrente quando l'utente desidera rileggere le istruzioni e lo restituisce al termine delle stesse. Rinunciando a quest'ultima possibilità si potrebbero dunque memorizzare quattro immagini e realizzare con apposita routinetta un discreto programma di animazione. Come è noto, l'immagine video assorbe 6912 byte; far posto in memoria a quattro di esse ha richiesto un drastico abbassamento di RAMTOP, come si vede alla linea 10. Inserito il programma rimangono liberi meno di 1200 byte, ma va detto che si possono conseguire ulteriori risparmi recuperando le zone cuscinetto tra le varie immagini in memoria.

I dati principali di lavoro sono costantemente visualizzati e aggiornati sulla riga 23. Vengono stampate le coordinate di PLOT, il modo di lavoro attivato fra i tre disponibili, l'inserimento della funzione di rientro della linea di disegno (RIEN), l'occupazione dei tre magazzini/memoria. Alcune scritte assumono il colore corrente di INK. La visualizzazione di questo display informativo sulla linea 23 ha consentito di lasciare all'utente l'intero schermo.

L'utilizzo più semplice del programma è senz'altro quello di produrre disegni fantasiosi. Con gli scolari io consiglio di abituarli a preparare prima il disegno su carta, con altezza uguale ai due terzi della base per rispettare le proporzioni dello Spectrum. Un uso più sofisticato, con miglior precisione e tempi più brevi di realizzazione, consiste nel riportare prima il disegno su un trasparente (ottimi quelli per lavagna luminosa, facendo uso degli appositi pennarelli o, meglio ancora, dell'inchiostro di china), fissare quest'ultimo allo schermo del televisore e procedere quindi al singolare "ricalcò". Il metodo si presta ottimamente per la riproduzione di cartine geografiche (e non dimenticate che lo Spectrum ha i colori).

Chi vuol caricare il programma nel minor tempo possibile può tralasciare le parti non fondamentali. Pertanto, dopo aver inserito il valore di RAMTOP con RANDOMIZE, inizi a battere il listato dalla linea 81. Si possono saltare le istruzioni dalla linea 1400 alla 1499 e la parte conclusiva dalla linea 9000 in poi. È comunque utile precauzione registrare il programma prima di farlo girare: non si sa mai.

Descrizione del programma

Linee

Commento

- 2** CF42II 190 che si legge dopo RANDOMIZE ha il solo scopo di presentare la copertina sempre con gli stessi colori ogni volta che il programma viene fatto girare. RAMPTOP è abbassata a 37399 per proteggere le immagini in memoria da indesiderate NEW.
- 3-10** Motivo grafico che compare sulla parte superiore sinistra dello schermo
- 30-70** Copertina; possibilità di leggere le istruzioni saltando alla linea 1400.
- 75** Rende esecutiva la linea 1400 in caso di risposta affermativa e inizializza la variabile j. Essa serve per il ritorno al programma principale dopo la lettura delle istruzioni, ritorno che può avvenire alla linea 80 o alla linea 130.
- 81** Questa POKE contiene il tempo di ritardo necessario prima che il tasto inizi a ripetersi. Inizialmente esso è di 35/50 di secondo. È stato portato a tre secondi per sicurezza, onde rendere inefficaci pressioni troppo prolungate sui tasti in fase di INPUT. Si è notato che la precauzione è utile soprattutto quando lavorano i bambini.
- 82** È la variabile di sistema che contiene il tempo di ritardo tra la ripetizione di un tasto e la successiva. Dall'iniziale 5 è stato elevato a 50 per i motivi detti in precedenza.
- 83** Questa variabile di sistema contiene la lunghezza del clic dei tasti. Inizialmente zero, il valore è stato posto a 35 perché sia più udibile quando un tasto viene premuto.
- 90** Definisce un carattere grafico associato al tasto "U". Si tratta di una freccia con punta in basso, usata in talune INPUT.
- 98** Questa matrice è usata per cambiare il colore di PAPER a richiesta dell'operatore.
- 100** Dopo l'inizializzazione delle variabili necessarie inizia il programma vero e proprio. Il modo RIENTRO è inattivo.
- 105** Prepara lo schermo.
- 110** a\$ contiene il display delle informazioni, in modo che l'operatore possa in ogni momento controllare i dati di lavoro fondamentali.
- 120** Posizione iniziale di PLOT a centro schermo.
- 130** Inizializza le funzioni OVER e INVERSE, come richiesto dal modo DISEGNO.
- 140** Stampa del display informativo sulla linea 23.
- 145** Stampa la condizione dei tre blocchi di memoria: "1", "2", o "3", se contengono un'immagine, "0", se i blocchi sono vuoti.
- 200** Tasto D. Mette il programma in modo DISEGNA, con le funzioni OVER e INVERSE non attivate. Aggiorna il display informativo stampando DISEGNA.
- 205** Tasto J. Consente di rileggere le informazioni.
- 210** Tasto C. Mette il programma in modo CANCELLA, con la funzione INVERSE attivata. Aggiorna il display informativo stampando CANCELLA.
- 215** Tasto X. Somma i due modi DISEGNA e CANCELLA, con OVER posta a 1. Pertanto il disegno viene creato dove lo schermo risulta libero, e cancellato se si passa sopra un disegno già esistente. Sul display viene stampato DIS/CANC.
- 220** Tasto V. Pulisce lo schermo con CLS e riporta il programma alla linea 100.
- 225** Tasto 9. Riproduce sulla parte destra dello schermo il disegno esistente sulla sinistra: a specchio. Quando si intende sfruttare questa possibilità, è logico che bisogna disegnare solo sulla sinistra, lasciando completamente libera la parte destra.
- 230** Tasto S. Consente di salvare il disegno, su nastro o stampante, come vedremo.
- 235** Tasto E. Consente di memorizzare l'immagine in uno dei tre magazzini/memoria predisposti. È appena il caso di dire che se inviate un'immagine dove già ce n'è un'altra, la seconda cancella e sostituisce la prima.
- 240** Tasto R. Richiama sullo schermo le immagini memorizzate.

- 245** Tasto H. Sposta il punto di PLOT a discrezione dell'utente.
- 250** Tasto I. Consente di cambiare il colore di INK.
- 255** Tasto K. Invia alla routine per colorare l'interno delle figure chiuse.
- 260** Tasto P. Consente di cambiare il colore di PAPER. Uso della DIM b\$(704) aperta alla linea 98.
- 265** Tasto G. Consente di stampare scritte sul disegno.
- 270** Tasto B. Cambia il colore di BORDER.
- 275** Tasto T. Disegna un triangolo equilatero.
- 280** Tasto O. Disegna un cerchio.
- 285** Tasto Q. Disegna un quadrato.
- 290** Tasto F. Cambia il valore logico della variabile "ri", mettendo in atto la funzione RIENTRO della linea di disegno quando è esclusa o disattivandola quando è in funzione. Il display informativo comunica il rientro con il messaggio RIEN.
- 295** Tasto L. Consente di tracciare linee dal punto di PLOT ad altro, le cui coordinate vanno introdotte con INPUT.
- 300** Tasti 5 e 8. Si disegna spostando la posizione di scrittura sull'ascissa, cioè sull'asse delle x.
- 310** Tasti 6 e 7. Si disegna spostando la posizione di scrittura sull'ordinata, cioè sull'asse delle y.
- 335** Tasto 1. Decrementa l'ascissa e incrementa l'ordinata. Viene tracciata una linea retta verso la parte sinistro-alta dello schermo, con angolo di 45° rispetto all'ascissa.
- 340** Tasto 0. Analogo all'1, ma con andamento della retta verso l'angolo superiore destro.
- 345** Tasto Z. Analogo ai precedenti, ma con andamento della retta verso la parte inferiore sinistra.
- 350** Tasto M. Analogo ai precedenti, ma con andamento della retta verso la parte inferiore destra.
- 360** Se la funzione di rientro della linea di disegno non è attivata (NOT ri), il programma va alla subroutine 2200 per il controllo delle coordinate di PLOT e l'eventuale messaggio di avvertimento "FUORI SCHERMO".
- 370-373** Se la funzione di RIENTRO è attiva, la linea di disegno, una volta raggiunto un lato dello schermo rientra dalla parte opposta. Si noti infatti che le coordinate di PLOT giunte al valore massimo vengono reinizializzate a quello minimo, e viceversa.
- 390** Stampa delle coordinate di PLOT.
- 400** Stampa del punto di PLOT.
- 500** Ritorna all'inizio del ciclo principale.
- 1000** Il disegno può essere salvato su nastro o stampante. Dopo il controllo sul tasto premuto (linea 1015), se viene risposto "nastro", bisogna dare un nome al disegno, con l'accortezza che non sia più lungo di 10 lettere, altrimenti lo Spectrum non lo accetta. La lunghezza viene comunque controllata con la linea 1035.
- 1100** Quando si vogliono tracciare linee dal punto di PLOT ad un altro, si devono fornire in INPUT le coordinate del punto ove la linea avrà termine. Per i distratti, in INPUT vengono rammentati i valori del punto plottato e quelli minimi e massimi per l'asse su cui si sta lavorando. Si sono usati i simboli < > per l'asse delle x e le frecce ↓ e ↑ per quello delle y. Come ulteriore sicurezza viene anche controllata la liceità del valore introdotto. Ribadisco che non bisogna dare l'incremento, o il decremento, delle coordinate così come richiesto da DRAW, ma i valori del punto ove la retta avrà termine. Provvede poi il programma con la linea 1140 ad eseguire i calcoli necessari. Prima di ritornare al ciclo di lavoro principale c'è la possibilità di cancellare la linea disegnata, rispondendo negativamente alla domanda di conferma. Se la retta viene confermata, il nuovo punto di PLOT viene spostato là ove essa termina, diversamente viene lasciato ai valori precedenti e la linea stessa cancellata.
- 1200** Controlla che i dati introdotti siano consentiti per i valori di PAPER, BORDER ed INK; diversamente segnala l'errore.
- 1300** Se lo si vuole, la posizione di PLOT può essere spostata, indicando nuovi valori. In INPUT vengono ricordati i valori correnti e gli estremi di quelli consentiti. Il controllo è effettuato solo sui valori massimi, perché è sempre possibile sbagliare e premere 6 invece di 5. Non c'è controllo per

E' IN EDICOLA

**LA PRIMA
E UNICA
RIVISTA DI
VIDEOGAMES - COMPUTER
GIOCHI ELETTRONICI**

Video Giochi



**GRUPPO
EDITORIALE
JACKSON**



Dalla grande edicola Jackson

Tutto sul personal computer

PC

L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.

10 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 40.000

Personal

L'unica rivista indipendente per gli utenti del personal computer Olivetti.

10 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.

9 numeri all'anno: L. 2.000 a numero
Abbonamento: solo L. 15.000

PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.

11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 34.000

Bit

La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato...

11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 43.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 Mi

Desidero ricevere GRATIS un numero della Rivista _____

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

valori inferiori a zero, perché l'inserimento di un numero negativo non può che essere intenzionale, comportando l'operazione di premere due tasti contemporaneamente (symbol shift + j).

Chi dovesse farlo manderà in blocco il programma ed io aggiungo che se lo sarà proprio meritato. È utile far notare che non sempre si sa stimare con esattezza il nuovo punto di PLOT. Ce lo ritroviamo dunque dove non è gradito: nessuna paura! Premete C, attivando il modo CANCELLA, chiamate ancora lo spostamento con H e introducete i nuovi valori. Dopo qualche tentativo avrete senz'altro la posizione desiderata.

1400

Contiene una breve routine in linguaggio macchina per memorizzare lo schermo e consentire la riletture delle istruzioni. I valori xx e yy sono l'indirizzo a partire dal quale verrà memorizzata l'immagine completa di attributi. In questo caso corrisponde a 37.400 e continua per le 6911 locazioni successive. La routine è posta a 65.500, cioè quasi al termine fisico della RAM (65535), e viene chiamata con la linea 1430.

1450

Ripulito lo schermo hanno inizio le istruzioni che terminano alla linea 1485.

1492

Viene inserita la routine per il richiamo dell'immagine di lavoro. Essa è posta dopo la precedente, cioè a partire da 65511 e viene chiamata con la linea 1497.

1499

Le istruzioni possono essere chiamate da due punti del programma: dalla linea 75 e dalla linea 205. Il ritorno deve quindi avvenire in due punti differenti. L'indirizzamento esatto è ottenuto tramite l'operatore logico NOT e la variabile j. Se essa è posta ugualmente a 1 (linea 75), il ritorno è alla linea 80; se posta a zero (linea 205), il ritorno è alla linea 130.

1500

La memorizzazione dell'immagine sullo schermo si ottiene premendo E e inserendo il numero corrispondente al blocco di memoria ove si desidera parcheggiarla. Il primo blocco ha inizio all'indirizzo 44.400, il secondo a 41.400, il terzo a 58.400. Come si può calcolare, l'intervallo è di 7K invece dei 6912 byte necessari (crepi l'avarizia!). La linea 1525 controlla che non venga premuto un tasto diverso da quelli indicati in INPUT. Ogni volta che un'immagine viene memorizzata la rispettiva variabile (m1, m2, m3) viene posta uguale alla cifra di m (1, 2 o 3) e visualizzata sul display di riga 23. La routine di memorizzazione viene inserita con le linee 1570-1580 e chiamata con la successiva. Essa è posta all'indirizzo 65.500 ed occupa 24 byte (12 per memorizzare e 12 per richiamare).

1600

Il richiamo avviene con il tasto "R" e successiva indicazione dell'immagine desiderata. È forse il caso di dire che una volta richiamata un'immagine il blocco che la conteneva non viene liberato, ma continua a conservare l'immagine fino a quando non verrà spento lo Spectrum. Nemmeno NEW la cancella, essendo la stessa posta oltre RAMTOP. Pertanto è forse più esatto dire che viene ricopiata sullo schermo.

1700

Coloritura interna di figure chiuse. Sempre pensando ai ragazzi quali utenti potenziali, si è inserita una linea che invita a controllare il punto di PLOT, confermando che esso è all'interno della figura. Con questa routine non è possibile colorare figure contorte o con angoli morti. Potrebbe rendersi necessario lo spostamento di PLOT e la chiamata della routine a più riprese.

1800

Questa routine consente di inserire scritte nel disegno. Per facilitare l'operazione, un'altra subroutine (linee 1850-1870) traccia sullo schermo un reticolato con passo di 8 pixel, visualizzando le 704 locazioni di stampa. Diventa semplice calcolare i dati richiesti relativi a riga e colonna. Dopo l'inserimento del testo il reticolo viene cancellato ed il testo stampato.

1900

La routine stampa un triangolo equilatero. Prima di chiedere il valore in pixel dell'altezza il programma calcola la lunghezza massima consentita per il lato in relazione al punto di PLOT (variabile x1). Indi calcola l'altezza necessaria per il valore trovato e, con la linea 1903, la confronta con la posizione di PLOT sull'ordinata. Dei due valori assu-



Dalla grande edicola Jackson

Tutto sull'hobby e home computer

strumenti MUSICALI

Il mondo delle 7 note in versione...elettronica. Con test strumentali, novità e analisi del mercato, servizi speciali. 10 numeri all'anno: L. 3.000 a numero Abbonamento: solo L. 24.000

VIDEO Giochi

La guida indiscussa al fantastico mondo dei videogames. La più eccitante, divertente, istruttiva rassegna del settore. 11 numeri all'anno: L. 3.500 a numero Abbonamento: solo L. 31.000

elektor

Il mensile di elettronica venduto in mezzo milione di copie e redatto in 7 lingue. Con articoli su: applicazioni, progettazioni, sperimentazioni, invenzioni. 11 numeri all'anno: L. 3.000 a numero Abbonamento: solo L. 29.000

hc

HOME COMPUTER
La rivista del computer in casa. "Prove su strada" di software e programmi per tutti i personal computer. 11 numeri all'anno: L. 3.500 a numero Abbonamento: solo L. 31.500

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 Mi

Desidero ricevere GRATIS un numero della Rivista _____

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

me il minore e ne informa l'operatore in INPUT. Introdotto il valore desiderato si controlla che esso non sia superiore al consentito, infine stampa il poligono con il punto di PLOT sul vertice superiore.

2000

Stampa del cerchio. Il programma calcola il massimo raggio ammesso, lo comunica in INPUT e controlla che il valore introdotto non sia superiore.

2100

Stessa procedura e identico controllo per il quadrato di cui va precisato il lato. Il poligono viene stampato in modo che il punto di PLOT si trovi a metà del lato superiore. Non è difficile modificare la routine per fare in modo che venga a trovarsi al centro della figura.

L'osservazione del listato permette di notare che triangolo, quadrato e cerchio vengono stampati usando una variabile per OVER e INVERSE. Infatti, normalmente la stampa avviene in modo DISEGNO, quindi con le due funzioni aventi valore zero. Se si desidera cancellare la figura subito dopo averla disegnata, è sufficiente chiamare il modo CANCELLA e fornire in INPUT il valore già dato per stamparla.

2200

Quando la funzione RIENTRO della linea di disegno non è attiva, questa routine controlla che il punto plottato non abbia coordinate eccedenti i valori massimi e minimi consentiti dal computer. Se necessario, avverte dell'errore, anche con una nota sonora.

2300

Specchio. Questa routine, più che una utilità, rappresenta una curiosità. Provvede a riprodurre in modo speculare sulla destra dello schermo la parte sinistra. Come si nota dal listato, vengono letti tutti i punti dell'ascissa per trovare quelli accesi e quindi accendere quelli speculari. Successivamente si incrementa l'ordinata e l'operazione viene ripetuta fino al limite superiore dello schermo. In BASIC occorrono diversi minuti per l'intera operazione.... e una certa pazienza. Vale come esempio di lettura dello schermo e può essere utile come esercizio di programmazione. Un altro utile esercizio può essere quello di modificare la routine, facendo riprodurre nella parte inferiore dello schermo ciò che compare in quella superiore. Particolare importante. Il disegno viene eseguito con il colore corrente di INK; successivamente le linee 2340-2370 provvedono al riporto degli attributi, andando a leggere le locazioni di memoria che li contengono.

Qualora la parte destra non fosse libera, l'eventuale disegno non verrebbe cancellato e quello riportato andrebbe a collocarsi sotto quello esistente, diciamo pure in secondo piano.

9000-9999

Linea di lancio. Routine di salvezza e di verifica. Calcolo della memoria rimasta libera.

I comandi a disposizione

D	entra in modo DISEGNA
C	entra in modo CANCELLA
X	entra in modo DISEGNA/CANCELLA
V	CLS
S	SAVE
E	memorizzazione immagini
R	richiamo immagini
G	stampa di testi
J	richiesta di istruzioni
H	spostamento del punto di PLOT
K	colorazione parte interna di figure chiuse
F	rientro della linea di scrittura
L	tracciamento di una retta
I	cambio di INK
P	cambio di PAPER
B	cambio di BORDER
T	disegna un triangolo
Q	disegna un quadrato
O	disegna un cerchio
5	disegna verso sinistra
8	disegna verso destra
6	disegna verso il basso
7	disegna verso l'alto



Fig. 1 - Immagine ottenuta utilizzando la Seikosha GP-505

1	disegna verso sinistra-alto
0	disegna verso destra-alto
Z	disegna verso sinistra-basso
M	disegna verso destra-basso
9	Funzione "specchio"

Descrizione delle variabili

bor	colore di BORDER
pap	colore di PAPER
ink	colore di INK
ov	valore di OVER
in	valore di INVERSE
ri	valore della funzione di RIENTRO LINEA
X, Y, X1, Y1	coordinate di disegno e tracciamento poligoni
n, m	variabili di servizio nei cicli FOR... NEXT
j	variabile di lavoro
m1, m2, m3	indicatori di memorizzazione immagini
a\$	display informativo con i dati di lavoro
b\$	DIM
va, i\$, h	valori in INPUT
xx, yy	indirizzo per memorizzare immagini

Tavolozza versione per ZX Spectrum 48K

```

1 REM ramtop e copertina
2 RANDOMIZE 190: CLEAR 37399:
  BORDER 0: PAPER 0: INK 6: CLS
3 FOR n=60 TO 175 STEP 5
4 PLOT 0,n: DRAW n,-n+175
5 INK INT (RND*7)+1: BEEP .04,n/4
6 NEXT n
10 DRAW 45,0
30 FOR n=0 TO 20 STEP 2: READ a$,m
31 BEEP .03,n
35 PRINT INK 1+INT (RND*6);AT n,m;
  a$
36 BEEP .05,50-n
37 READ a$,m: PRINT INK 1+INT (
  RND*6);AT n+1,m;a$
40 NEXT n
45 DATA "{3SG8}",29,"{SG8} {SG8}",2
  9,"*** {3SG8}",25,"* {SG8}
  {SG8}",27,"{3SG8} * {SG8}
  {SG8}",22
50 DATA "{SG8}*",24,"*** {SG8} ***
  ",18,"* * {SG8}",18,"{SG8} * *
  {3SG8}",16,"{SG8} * *",16
55 DATA "*** {SG8} ***",12,"* *
  {SG8}",12,"{SG8} {SG8} * *
  {3SG8}",8,"{SG8} {SG8} * **",8,"*
  ** {SG8} {SG8} ***",4
60 DATA "* * {SG8} {SG8}",4,"
  {3SG8} *** {SG8}",0,"{SG8} * *
  ",1,"{SG8} * **",1,"{SG8}",1,"
  {SG8}",1
65 DATA "premi J per le istruzioni"
  ,7
67 LET pap=0: LET ink=7
70 PAUSE 0
75 IF INKEY$="j" THEN LET j=1:
  GO TO 1400
78 GO TO 81
80 PRINT PAPER 6; INK 0;AT 21,7;"P
  REMI UN TASTO QUALSIASI.":
  PAUSE 0
81 POKE 23561,150
82 POKE 23562,50
83 POKE 23609,35
90 REM caratteri grafici
91 RESTORE 96
93 FOR n=0 TO 7: READ a
95 POKE USR "u"+n,a: NEXT n
96 DATA 0,8,8,8,42,28,8,0
97 REM inizializza variabili
98 DIM b$(704)
99 LET m1=0: LET m2=0: LET m3=0
100 LET ri=0: LET bor=0
105 BORDER bor: PAPER pap: INK ink:
  OVER 0: INVERSE 0: CLS
110 LET a$="X= {SG8}Y= {SG8}DISE
  GNA{7SG8}MEM "
120 LET x=127: LET y=87
130 LET ov=0: LET in=0
140 PRINT #0; INK ink;AT 1,0;a$
145 PRINT #0;AT 1,29;m1;m2;m3
200 IF INKEY$="d" THEN LET ov=0:
  LET in=0: PRINT #0;AT 1,12;"DIS
  EGNA{G8}"
205 IF INKEY$="j" THEN LET j=0:
  GO TO 1400
210 IF INKEY$="c" THEN LET in=1:
  LET ov=0: PRINT #0;AT 1,12;"CAN
  CELLA"
215 IF INKEY$="x" THEN LET in=0:
  LET ov=1: PRINT #0;AT 1,12;"DIS
  /CANC"
220 IF INKEY$="v" THEN GO TO 100
225 IF INKEY$="9" THEN GO TO 2300
230 IF INKEY$="s" THEN GO TO 1000
235 IF INKEY$="e" THEN GO TO 1500

```

```

240 IF INKEY$="r" THEN GO TO 1600
245 IF INKEY$="h" THEN GO TO 1300
250 IF INKEY$="i" THEN INPUT
    INVERSE 1;"Nuovo INK (0/7) ";i$
    : GO SUB 1200: LET ink=VAL i$:
    INK ink: PRINT #0; INK ink;AT 1
    ,0;a$: GO TO 390
255 IF INKEY$="k" THEN GO TO 1700
260 IF INKEY$="p" THEN INPUT
    INVERSE 1;"Nuovo PAPER (0...7)
";i$: GO SUB 1200: PRINT
    PAPER VAL i$; OVER 1;AT 0,0;b$(
    TO 704): PRINT #0;AT 1,0;a$:
    GO TO 390
265 IF INKEY$="g" THEN GO TO 1800
270 IF INKEY$="b" THEN INPUT
    INVERSE 1;"Nuovo BORDER (0...7)
";i$: GO SUB 1200: BORDER VAL i
$: PRINT #0; INK ink;AT 1,0;a$:
    GO TO 390
275 IF INKEY$="t" THEN GO TO 1900
280 IF INKEY$="o" THEN GO TO 2000
285 IF INKEY$="q" THEN GO TO 2100
290 IF INKEY$="f" THEN LET ri=NOT r
i: PRINT #0; INK ink;AT 1,21;("R
IEN" AND ri)+("{4SG8}" AND NOT r
i)
295 IF INKEY$="l" THEN GO TO 1100
300 LET x=x+(INKEY$="8")-(INKEY$="5"
)
310 LET y=y+(INKEY$="7")-(INKEY$="6"
)
335 IF INKEY$="l" THEN LET x=x-1:
    LET y=y+1
340 IF INKEY$="0" THEN LET x=x+1:
    LET y=y+1
345 IF INKEY$="z" THEN LET x=x-1:
    LET y=y-1
350 IF INKEY$="m" THEN LET x=x+1:
    LET y=y-1
360 IF NOT ri THEN GO SUB 2200:
    GO TO 390
370 IF ri THEN IF x<0 THEN LET x=2
55
371 IF ri THEN IF x>255 THEN LET x
=0: PRINT #0; INK ink;AT 1,3;"
{2SG8}"
372 IF ri THEN IF y<0 THEN LET y=1
75
373 IF ri THEN IF y>175 THEN LET y
=0: PRINT #0; INK ink;AT 1,9;"
{2SG8}"
390 PRINT #0; PAPER 9;AT 1,2;x;"
{SG8}";AT 1,8;y;"{SG8}"
400 PLOT OVER ov; INVERSE in;x,y
500 GO TO 145
1000 REM funzione SAVE
1010 INPUT "Nastro o Stam
pante?(N/S)";i$
1015 IF i$<>"n" AND i$<>"s"
    THEN BEEP 1,-24: GO TO 1101
1020 IF i$="s" THEN COPY : GO TO 130

1030 INPUT "Nome(max 10 lett) ";i$
1035 IF LEN i$>10 THEN BEEP 1,-24:
    GO TO 1030

```

```

1036 IF i$="" THEN GO TO 1030
1040 SAVE i$SCREEN$
1050 GO TO 130
1100 REM funzione linea
1105 INPUT INVERSE 1;" X da ";(x)
;" a <0/255>";i$
1107 IF i$="" THEN GO TO 1100
1110 IF VAL i$>255 THEN BEEP 1,-24:
    GO TO 1105
1115 LET x1=VAL i$
1120 INPUT INVERSE 1;" Y da ";(y)
;" a U0/175↑";i$
1122 IF i$="" THEN GO TO 1120
1125 IF VAL i$>175 THEN BEEP 1,-24:
    GO TO 1120
1130 LET y1=VAL i$
1140 DRAW OVER 0;x1-x,y1-y
1150 INPUT INVERSE 1;"Confermi?(S/N)
";i$: IF i$="s" THEN LET x=x1:
    LET y=y1: GO TO 130
1155 PLOT OVER 1;x,y: DRAW OVER 1;x
1-x,y1-y
1160 GO TO 130
1200 REM controllo DATI
1201 IF i$="" THEN GO TO 1210
1205 IF CODE i$>47 AND CODE i$<56
    THEN RETURN
1210 PRINT #0; PAPER 6; INK 0;"DATO E
RRATO/RICHIAMA LA FUNZIONE":
    BEEP 4,-24: INPUT ""
1220 GO TO 130
1300 REM sposta PLOT
1310 INPUT INVERSE 1;" X da ";(
x);" a <0/255> ";i$
1312 IF i$="" THEN GO TO 1300
1315 IF VAL i$>255 THEN BEEP 1,-24:
    GO TO 1310
1320 LET x=VAL i$
1325 INPUT INVERSE 1;" Y da ";(y)
;" a U0/175↑";i$
1327 IF i$="" THEN GO TO 1325
1330 IF VAL i$>175 THEN BEEP 1,-24:
    GO TO 1325
1335 LET y=VAL i$
1350 GO TO 130
1400 REM stampa istruzioni
1402 LET ov=0: LET in=0
1405 RESTORE 1425
1410 LET xx=170: LET yy=146
1415 FOR n=0 TO 11: READ a
1420 POKE 65500+n,a: NEXT n
1425 DATA 33,00,64,17,xx,yy,1,00,27,2
37,176,201
1430 RANDOMIZE USR 65500
1450 INVERSE 0: PAPER 7: INK 0: CLS
1455 PRINT PAPER 6;"ISTRUZIONI ISTRU
ZIONI ISTRUZIONI{32SG3}"
1458 PRINT " Questo programma consen
te di disegnare sullo schermo
ed offrediverse possibilita'.""
- Si puo' cancellare"" - sovra
pporre"" - Salvare il disegno"
" - Richiamare le istruzioni
senza perdere il disegno che
viene ristampato alla fine
della lettura."

```

LOAD**Tavolozza**

```

1460 PRINT " - Memorizzare fino a tre
      dise- gni e, se necessario r
      ichia- marli. Le tre zone di
      memo- ria occupate sono segn
      alate con MEM seguito da 1,
      2, 3."
1462 PRINT " - Far rientrare dalla pa
      rte op- posta dello schermo la
      linea del disegno. La funzio
      ne vien segnalata con RIEN"
1465 PRINT #0;AT 1,9;"premi un tasto"
      : PAUSE 0: INPUT ""
1468 PAPER 0: INK 0: CLS : PRINT
      PAPER 6;"TASTO E FUNZIONE CORRI
      SPONDEnte."
1470 PRINT PAPER 7' BRIGHT 1;"D";
      BRIGHT 0;" disegna"" BRIGHT 1;
      "C"; BRIGHT 0;" cancella""
      BRIGHT 1;"X"; BRIGHT 0;" sovrappone
      (DIS/CANC)"" BRIGHT 1;"V";
      BRIGHT 0;" cancella schermo: CL
      S"" BRIGHT 1;"S"; BRIGHT 0;" sal
      va su nastro o stampante"
1472 PRINT PAPER 7' BRIGHT 1;"J";
      BRIGHT 0;" richiama le istruzio
      ni"" BRIGHT 1;"E"; BRIGHT 0;" m
      emorizza il disegno"" BRIGHT 1;
      "R"; BRIGHT 0;" richiama dalla m
      emoria"
1474 PRINT PAPER 7' BRIGHT 1;"K";
      BRIGHT 0;" colora interno figur
      e chiuse"" BRIGHT 1;"H";
      BRIGHT 0;" sposta punto di PLOT
      ": PRINT #0;AT 1,9;"premi un tas
      to": PAUSE 0: INPUT ""
1475 CLS : PRINT PAPER 7; BRIGHT 1;"
      I"; BRIGHT 0;" cambia INK""
      BRIGHT 1;"P"; BRIGHT 0;" cambia
      PAPER"" BRIGHT 1;"B"; BRIGHT 0
      ;" cambia BORDER"
1477 PRINT ' PAPER 7; BRIGHT 1;"G";
      BRIGHT 0;" stampa parole""
      BRIGHT 1;"L"; BRIGHT 0;" tracci
      a linee dando coordinate"
1478 PRINT ' PAPER 7; BRIGHT 1;"T";
      BRIGHT 0;" disegna un triangolo
      equilat."" BRIGHT 1;"O";
      BRIGHT 0;" disegna un cerchio"
      ' BRIGHT 1;"Q"; BRIGHT 0;" diseg
      na un quadrato"" BRIGHT 1;"F";
      BRIGHT 0;" rientro della linea
      di disegno"
1479 PRINT ' PAPER 7; BRIGHT 1;"9";
      BRIGHT 0;" copia a destra la pa
      rte sini-";TAB 2;"stra dello sch
      ermo; a specchio": PRINT INK 7;
      AT 21,9;"premi un tasto":
      PAUSE 0: CLS
1480 PRINT PAPER 6; INK 0;"DISEGNA U
      SANDO I SEGUENTI TASTI."
1481 INK 7: PLOT 96,76: DRAW 72,0:
      PLOT 132,40: DRAW 0,72: PLOT 96
      ,40: DRAW 72,72: PLOT 168,40:
      DRAW -72,72
1482 INK 0: PRINT PAPER 6; INK 0;
      AT 6,10;"1";AT 6,16;"7";AT 6,22;
      "0"
1483 PRINT PAPER 6;AT 12,10;"5";AT 1
      2,22;"8";AT 18,10;"Z";AT 18,16;"
      6";AT 18,22;"M"
1485 PRINT #0;AT 1,9;"premi un tasto"
      : PAUSE 0: INPUT ""
1490 RESTORE 1495
1492 FOR n=0 TO 11: READ a
1494 POKE 65512+n,a: NEXT n
1495 DATA 33,xx,yy,17,00,64,1,00,27,2
      37,176,201
1498 OVER ov: INVERSE in: CLS :
      RANDOMIZE USR 65512
1499 GO TO (130 AND NOT j)+(80 AND j)
1500 REM memorizzazione immagini
1502 RESTORE 1580
1505 INPUT INVERSE 1;"Quale memori
      a?(1-2-3) ";i$
1510 IF i$="1" THEN GO TO 1530
1515 IF i$="2" THEN GO TO 1540
1520 IF i$="3" THEN GO TO 1560
1525 BEEP 1,-24: GO TO 1505
1530 LET m1=1: LET xx=30: LET yy=174
1535 GO TO 1570
1540 LET m2=2: LET xx=145: LET yy=201
1545 GO TO 1570
1560 LET m3=3: LET xx=5: LET yy=229
1570 FOR n=0 TO 11: READ a
1575 POKE 65500+n,a: NEXT n
1580 DATA 33,00,64,17,xx,yy,1,00,27,2
      37,176,201
1590 RANDOMIZE USR 65500
1595 PRINT #0;AT 0,0; PAPER 6; INK 0;
      " MEMORIZZAZIONE EFFETTUATA.
      Premi un tasto per continuare.
      ": PAUSE 0: INPUT "" : GO TO 130
1600 REM richiamo immagini
1605 RESTORE 1680
1610 INPUT INVERSE 1;"Quale schermo
      vuoi? (1-2-3)";i$
1615 IF i$="1" THEN GO TO 1630
1620 IF i$="2" THEN GO TO 1640
1625 IF i$="3" THEN GO TO 1650
1627 BEEP 1,-24: GO TO 1610
1630 LET xx=30: LET yy=174
1635 GO TO 1670
1640 LET xx=145: LET yy=201
1645 GO TO 1670
1650 LET xx=5: LET yy=229
1670 FOR n=0 TO 11: READ a
1675 POKE 65512+n,a: NEXT n
1680 DATA 33,xx,yy,17,00,64,1,00,27,2
      37,176,201

```



```

1690 RANDOMIZE USR 65512
1695 GO TO 130
1700 REM colora interno figure
1710 INPUT INVERSE 1;"PLOT e' nella
figura?(S/N)";i$
1720 IF CODE i$<>115 THEN BEEP 1,-
24: GO TO 130
1730 FOR n=x TO 255: FOR m=y TO 175
1740 PLOT n,m
1750 IF POINT (n,m+1)=0 THEN NEXT m
1755 IF POINT (n+1,y)=0 THEN NEXT n
1760 FOR n=x TO 255: FOR m=y-1 TO 0
STEP -1
1762 PLOT n,m
1765 IF POINT (n,m-1)=0 THEN NEXT m
1770 IF POINT (n+1,y-1)=0 THEN
NEXT n
1780 FOR n=x-1 TO 0 STEP -1: FOR m=y-
1 TO 0 STEP -1
1783 PLOT n,m
1785 IF POINT (n,m-1)=0 THEN NEXT m
1788 IF POINT (n-1,y-1)=0 THEN
NEXT n
1790 FOR n=x-1 TO 0 STEP -1: FOR m=y
TO 175
1792 PLOT n,m
1794 IF POINT (n,m+1)=0 THEN NEXT m
1795 IF POINT (n-1,y)=0 THEN NEXT n
1798 GO TO 130
1800 REM stampa scritte
1810 GO SUB 1850
1815 INPUT INVERSE 1;"Riga=t0/21
U";va; INVERSE 1;"Colonna=
<0/31>";h
1820 IF va>21 OR h>31 THEN BEEP 1,-2
4: GO TO 1815
1825 INPUT INVERSE 1;"Scritta"
;i$
1827 GO SUB 1850
1830 PRINT PAPER 8; INK 8;AT va,h;i$

1835 GO TO 130
1850 FOR n=0 TO 255 STEP 8
1855 PLOT OVER 1;n,0: DRAW OVER 1;0
,175: NEXT n
1860 FOR n=0 TO 175 STEP 8
1865 PLOT OVER 1;0,n: DRAW OVER 1
;255,0: NEXT n
1870 RETURN
1900 REM triangolo
1901 LET x1=(2*x AND x<=127)+(2*(255-
x) AND x>127)
1902 LET x1=INT SQR (x1+2-(x1/2)+2)
1903 LET x1=(x1 AND x1<=y)+(y AND x1>
y)
1905 INPUT INVERSE 1;"Altezza? (max
";(x1);)"";va: IF va>x1 THEN
BEEP 1,-24: GO TO 1905
1910 LET x1=INT (va/SQR 3)
1915 OVER ov: INVERSE in
1920 DRAW -x1,-va: DRAW 2*x1,0:
DRAW -x1,va
1930 GO TO 130
2000 REM cerchio
2001 LET x1=(x AND x<=127)+(255-x
AND x>127)

```

```

2002 LET y1=(y AND y<=87)+(175-y
AND y>87)
2003 LET x1=(x1 AND x1<=y1)+(y1 AND x
1>y1)
2010 INPUT INVERSE 1;"Raggio (max ";
(x1);)" ";va: IF va>x1 THEN
GO TO 1210
2020 CIRCLE OVER ov; INVERSE in;x,y,
va
2030 GO TO 130
2100 REM quadrato
2101 LET x1=(2*x AND x<=127)+(2*(255-
x) AND x>127)
2102 LET x1=(x1 AND x1<=y)+(y AND x1>
y)
2108 INPUT INVERSE 1;"Lato (max ";(x
1);)" ";va
2110 IF va>x1 THEN GO TO 1210
2115 OVER ov: INVERSE in
2120 PLOT x-va/2,y: DRAW 0,-va:
DRAW va,0: DRAW 0,va: DRAW -va,
0
2130 GO TO 130
2200 REM controllo PLOT
2201 IF x>=0 AND y>=0 AND x<=255
AND y<=175 THEN RETURN
2205 LET x=x-(1 AND x>255)+(1 AND x<0
)
2210 LET y=y-(1 AND y>175)+(1 AND y<0
)
2220 PRINT #0;AT 1,0;"< STAI USCENDO
DALLO SCHERMO! >": BEEP 4,-24:
GO TO 130
2300 REM specchio
2305 PRINT #0;AT 0,0;"Copio a destra
la parte sinistradello schermo.
Pazienta 4 minuti"
2310 FOR n=0 TO 175: FOR m=0 TO 127
2320 IF POINT (m,n)=1 THEN PLOT 255-
m,n
2330 NEXT m: NEXT n
2340 LET va=22528: LET h=22544
2350 FOR n=1 TO 22: FOR m=15 TO 0
STEP -1
2355 POKE h+(15-m),PEEK (va+m)
2360 NEXT m
2365 LET va=va+32: LET h=h+32
2370 NEXT n
2375 BEEP 1,12: INPUT ""
2380 GO TO 130
9000 REM lancio e SAVE programma
9001 CLS : PRINT FLASH 1;AT 10,0;"
ferma il nastro "
9010 PRINT AT 21,0;"Premi un tasto pe
r incominciare.": PAUSE 0: RUN
9990 SAVE "tavolozza" LINE 9000
9991 CLS : PRINT "SAVE conclusa""'p
repara il nastro per la VERIFY e
premi PLAY"
9992 VERIFY "tavolozza"
9993 PRINT ""Programma O.K.""'CIAO
!"
9995 PRINT ""Memoria libera; bytes
";65535-USR 7962
9999 STOP

```

PRINT

**Recensioni
Libri**

I libri italiani vengono offerti con il 10% di sconto sul prezzo di copertina

di **Lucio Bragagnolo**

Ai confini dello Spectrum - applicazioni avanzate

di **P. Williams**
ed. **Gruppo Editoriale Jackson**



Altra novità Jackson per questa puntata di PRINT, più consueta come genere della precedente, ma ugualmente degna di segnalazione.

Si tratta del "solito" libro di programmi, comprendente una trentina di listati variamente distribuiti tra utility, giochi e curiosità. Comunque, come da moltissimo tempo a questa parte la Jackson ci ha abituato nel campo dei libri per Spectrum, anche questa volta l'accento non è messo sulla quantità del materiale (in fondo, come viene riconosciuto anche nell'introduzione, sono già stati pubblicati libri contenenti più di 70 programmi), bensì sulla qualità. Difatti, tutti i programmi inclusi sono abbondantemente commentati, descritti, analizzati, alcuni anche sotto forma di chiarissimo pseudocodice, così da costituire, più

I libri possono essere ordinati utilizzando il coupon a pag. 82

che una mera raccolta di listati, una nastroteca di consigli, routine e trucchi preziosi per tutti i Sinclairisti amanti della programmazione (moltissimi rispetto al totale degli utenti, contrariamente - ad esempio - a quanto accade in casa Commodore o accadrà in casa MSX). Diciamo nastroteca, in quanto al libro è acclusa una cassetta contenente tutti i programmi già pronti per essere utilizzati.

Le considerazioni di cui sopra giustificano alcune scelte dell'autore di "Ai confini dello Spectrum", quali lo scarsissimo uso del linguaggio macchina, anche in giochi bisognosi di velocità, o l'uso frequentissimo di REM, molto utili per capire il funzionamento del programma. Un esempio è dato dalla presenza di un programma di scacchi scritto interamente in BASIC: lentissimo nell'analisi delle mosse, è anche estremamente debole (cosa che, del resto, l'autore ammette onestamente nelle istruzioni): ciò non toglie che potrà servire da stimolo per tutti coloro che vorranno lavorarci sopra, migliorando e arricchendo il programma stesso: un domani potremo vedere arrivare in redazione qualche scacchista elettronico, scritto in BASIC, ma non troppo lento, e dotato di una discreta abilità, lontano parente del suo progenitore jacksoniano... ma non divaghiamo eccessivamente, e passiamo invece a un esame ravvicinato del libro. Innanzitutto notiamo l'ottima presentazione grafica, semplice, ma tutt'altro che sciatta, di estrema pulizia e chiarezza. I listati sono ottenuti con una stampante a margherita, così da consentire una perfetta leggibilità. È stata inoltre usata una speciale codifica dei caratteri grafici e degli spazi, atta a sgombrare il campo da qualunque equivoco di interpretazione dei listati. I programmi, dicevamo, sono di tutti i generi, e appaiono in ordine di complessità crescente, tanto che il primo listato che vale la

pena di prendere in considerazione è il quarto, "Orologio Spectrum", che visualizza sullo schermo un orologio analogico con tanto di sveglia e suoneria. Seguono un tradizionale, ma ineccepibile labirinto tridimensionale e un programmino intitolato "Test di geometria": quest'ultimo formula domande del tipo "Qual è il volume di una sfera con raggio 2?"; l'aspetto interessante è che viene accettato come risposta non solo il risultato esatto, ma anche una formula equivalente. Saltando il blackjack e una slot machine molto lenta, ma con grafica favolosa, arriviamo a un programma di "bubble sort", capace di visualizzare con grande simpatia ed efficacia la procedura seguita per ordinare 10 numeri, dal più grande al più piccolo, per mezzo di un piccolo carrello elevatore che, letteralmente, solleva i vari numeri per depositarli docilmente nella giusta posizione. Le versioni del programma sono tre, progressivamente più potenti: la terza è costruita in forma di subroutine, per poter essere inserita nel corpo di un altro programma, ed ha una velocità davvero sorprendente.

Segue un programma che calcola - e visualizza - il punto di intersezione nel piano cartesiano di due rette qualsiasi; dopo di che saltiamo al più tradizionale dei giochi, vale a dire l'immortale "Spectrum invaders", caratterizzato in questo libro dall'ottimo output grafico: gli alieni non sono astronavi, ma assomigliano vagamente a Orazio, il protagonista di "Hungry Horace" e "Horace goes to skiing"; il risultato è davvero impressionante (e nonostante l'uso del BASIC la velocità è accettabile).

Sempre pescando qua e là, avvicinandosi alla fine del libro si possono trovare cose piuttosto interessanti: le consuete, ma sempre utili, routine di renumber e cancellazione di blocchi di linee, un insolito programma di monitor, davvero ben fatto, e un trittico di listati intitolati "Analisi delle vendite", "Libro paga" e "Valutazione dei beni", che sapranno essere d'aiuto per la comprensione dei principi di funzionamento di un database.

Torniamo agli scacchi: il programma è di dimensioni sorprendentemente ridotte e, come dicevamo più sopra, è lento oltre che "stupido" (abbiamo giocato una partita sola, vinta in cinque mosse). Tuttavia è realizzato molto bene dal punto di vista grafico, e l'algoritmo di decisione di ogni mossa, per quanto allo stato embrionale, potrà servire da pun-

to di partenza per chi si senta in grado di cimentarsi nella fatica di migliorarlo e potenziarlo. Sempre riguardo agli scacchi, nel programma vi sono alcuni bug marginali, ma che vanno indicati: in primo luogo la scacchiera è ruotata di novanta gradi rispetto alla posizione corretta (che vuole una casella bianca nell'angolo in basso a destra della scacchiera); secondo, contrariamente al solito, il giocatore bianco è situato nella parte alta del piano di gioco; terzo, il computer pensa ugualmente alla sua mossa prima di verificare la presenza di uno scacco matto, così che prima di ammettere la sua sconfitta (o verificare la sua vittoria) muove, alterando la situazione. È chiaro che questo comportamento potrebbe in certi casi creare delle ambiguità piuttosto gravi.

Non ci sono invece bug di alcun genere nella dama, di livello superiore agli scacchi, ma che comunque non vi consigliamo come avversario preferito. Vale anche qui il discorso delle migliori già fatto in precedenza. Da notare che questo è uno dei pochi giochi facenti parziale ricorso al linguaggio macchina. Il libro si conclude con un'avventura di discreta complessità e tutto sommato ben realizzata, pur rimanendo, sia chiaro, un esempio e non un gioco su cui passare dei mesi. Ad ogni buon conto le ultime pagine del libro ospitano una serie di consigli, che favoriranno il raggiungimento della soluzione anche ai più ostinati.

Che dire ancora di "Ai confini dello Spectrum"? Un buon libro (completo, vale la pena di ricordarlo, di cassetta) che vale senz'altro più del prezzo di copertina e che vi consigliamo.

Il BASIC in 30 ore per lo Spectrum

di **Clive Prigmore**
ed. **Gruppo Editoriale Jackson**



Dopo cento raccolte di giochi, mille trattati sul linguaggio macchina, diecimila cosiddetti aiuti alla programmazione sembra finalmente che la fetta di editoria informatica dedicata allo Spectrum sia avviata a elevare notevolmente il proprio livello qualitativo. Lo spunto per questa ipotesi ci viene dato da "Il BASIC in 30 ore per lo Spectrum", autentico best seller nel Regno Unito (più di 100.000 copie vendute in Inghilterra, recita la copertina), sicuramente destinato a riscuotere successo anche sul mercato italiano, meno vivo, ma tutt'altro che saturo.

Il segreto, se così si può chiamare, dell'opera consiste nell'ennesimo uovo di Colombo: scrivere un libro che insegni a programmare... scrivendo un libro che insegni realmente a programmare.

Cosa vuol dire? Cerchiamo di spiegarci meglio. Ciò che intendiamo sottolineare è che finalmente siamo di fronte a un sussidio didattico vero e proprio, scritto, impaginato, disegnato, progettato a scopo di insegnamento, dove prima la maggioranza dei libri puntava soprattutto a dare il massimo di informazione nel minimo di pagine, col risultato di sconcertare il lettore, letteralmente sorpassato, dopo i primi capitoli introduttivi, dal ritmo frenetico del libro.

Beninteso, non si vuole negare l'utilità di tutte le pubblicazioni apparse finora; vogliamo però sottolineare che per un neofita l'approccio più produttivo è quello usato in "Il BASIC...".

Sfogliandolo distrattamente, l'aspetto è simile a quello di un libro di testo per le scuole: periodi evidenziati in rosso, numerosissime illustrazioni chiarificatrici, piccoli listati presenti dappertutto (anche qui le linee di programma significative sono visualizzate in rosso), con una differenza: la mancanza di testi lunghissimi e verbosi. La sintesi prevale sulla logorrea, in omaggio al vecchio detto inglese secondo cui chi impiega sei parole potendone usare anche solo cinque è capace di qualsiasi delitto. Colpisce anche la spaziatura tra le varie informazioni, non riempitivo per mancanza di idee quanto aiuto alla chiarezza: chi legge deve avere il tempo di "digerire" lo scritto e meditarci sopra.

Approfondendo l'esame, possiamo ora vedere quali sono le direttrici secondo le quali l'autore ha impostato il libro. Come leggiamo dall'introduzione, per saper utilizzare al meglio un personal computer bisogna conoscere per prima cosa il suo dialetto BASIC e, in seguito, imparare a realizzare buone strutture di programma. Il resto (ad esempio la tastiera) viene dopo, e difatti la gran parte del libro si occupa dei primi due punti. Altra cosa: il corso può essere seguito anche senza possedere un computer! Vi sono infatti esempi ed esercizi riservati agli utenti Spectrum, ed altri accessibili a chiunque. Infine, non si parla di hardware: tutto ciò che bisogna sapere per programmare un calcolatore è la manie-

ra migliore per fornirgli un problema così che esso risponda come desiderato: il resto, almeno finché si impara, non ha eccessiva importanza. Aggiungiamo a questo quadro di insieme la presenza di numerosi test, esercizi, verifiche, e sarà già possibile a questo punto farsi un'idea abbastanza precisa dell'insieme. Scendendo ancora maggiormente nei particolari, vediamo quali sono gli argomenti trattati. È importante sottolineare come nel libro non venga trattato l'intero BASIC Spectrum: ci sono istruzioni (OUT, POKE, USR...) e argomenti (variabili di sistema, linguaggio macchina, routine della ROM...) che non sono di utilità per chi impara a programmare e che anzi possono condurre a una pericolosa confusione mentale.

I capitoli del libro sono 9: il primo di essi è dedicato all'analisi di un elementare programma di somma di due numeri, in cui vengono introdotti LIST, SAVE, LOAD e VERIFY. Il capitolo 2 si intitola "Prendere le decisioni", e in esso si parla di IF... THEN, della logica booleana (illustrata molto chiaramente), di algoritmi, flowchart e dell'uso di GO TO, da non esasperare. Si prosegue con stringhe e liste, argomenti che consentono la trattazione di temi come variabili, separatori, file, READ... DATA, FOR... NEXT, ordinamento dell'output sullo schermo e altro ancora. Non manca neppure una parte dedicata allo slicing, completa della traduzione degli operatori RIGHT\$, MID\$ e LEFT\$ propri del BASIC Microsoft; si parla anche di sicure negli input, argomento che (esperienza redazionale) probabilmente pochissimi possono permettersi di ignorare.

Il capitolo 6 ("Soprattutto sui dadi e giochi") tratta di casualità (RND) e operazioni su stringhe, mentre nel capitolo 7 ("Grafica, suono e colore") l'attenzione è puntata sulle doti grafiche dello Spectrum, con in più una semplice spiegazione della teoria dei colori e del perché, ad esempio, il codice numerico del colore blu è proprio 1 e non, per dire, 3 o 47 (chi lo sa non suggerisca, please). Si parla anche di BEEP, anche se ovviamente non vi è moltissimo da dire in proposito.

L'argomento dell'ottavo capitolo è abbastanza insolito: l'elaborazione numerica (media aritmetica, frequenze, tabelle, i cosiddetti "dry run", problemi di rappresentazione dei numeri), mentre il capitolo 9 è una "introduzione all'elaborazione dati". Breve, ma succoso condensa in poche pagine nozioni utilissime sui vari sistemi di ricerca e ordinamento di un file, su come stendere un menu di programma e come organizzare un output su stampante. In conclusione, speriamo di essere riusciti a illustrare soddisfacentemente la "diversità" di questo libro rispetto agli altri: dovendo sintetizzare, è l'unico che fino ad oggi ci abbia fatto gustare il vero e proprio piacere di imparare.

ZX CLUB

I Club
Sinclair

Le notizie di ZX Club

IL GRUPPO UTILIZZATORI COMPUTER SINCLAIR NAPOLI ha recentemente esteso il suo campo di attività ai computer con standard MSX (Sony, Philips, Spectravideo...). Se, come non dubitiamo, la competenza e l'impegno saranno gli stessi di quelli profusi in ambito Spectrum, il club diventerà sicuramente un apprezzato punto di riferimento anche per gli utenti MSX! Scrivete o telefonate per informazioni al

GRUPPO UTILIZZATORI COMPUTER - Napoli
LIBERA ASSOCIAZIONE DI UTENTI DI COMPUTER

sinclair & **MSX**

SOCIO PROGRAMMATORE

INFORMAZIONI ED ISCRIZIONI TELEFONANDO ALLO 081/617368 - 7623121



Gruppo Utilizzatori Computer Sinclair e MSX Napoli
c/o Roberto Chimenti
via Luigi Rizzo, 18
80124 NAPOLI

A quanto ci risulta il numero telefonico è sempre 081/61.73.68; tuttavia c'è la possibilità che sia cambiato in questi ultimissimi giorni, diventando 081/76.23.121.



Gli annunci di ZX Club

Cerco possessori presenti e futuri di QL per scambiare software e idee, e fondare un "QL user Club". Inoltre svendo software e manuali originali per Spectrum. Scrivere a Roberto GHEZZI via Volontari del Sangue, 202 20099 SESTO SAN GIOVANNI (MI)

È nato un SINCLAIR CLUB per utenti Spectrum! Quota di iscrizione Lit. 10.000, e prezzi minimi per programmi. Oltre 700 titoli. Luigi FABERI via Campagnola, 3 25079 VOBARNO (BS) Tel. 0365/61.83.88

1000 programmi per Commodore 64 e Spectrum ti aspettano! Inviare Lit. 1.000 per catalogo completo a COMPUTER CLUB Casella Postale 7220 16166 GENOVA - QUINTO (GE)

I programmi di ZX Club

Ricordate "Il test del vostro futuro"? Ebbene, il SINCLAIR CLUB ROMA - TALENTI ha colpito ancora, presentandoci "Test 2 - siete zingari o pantofolai?". A differenza di quanto accade in campo cinematografico, questa volta il secondo episodio è migliore del primo. Il programma è sempre per lo Spectrum 48K: i "sedicikappisti" possono comunque lavorare sul listato e adattarlo alla loro disponibilità di memoria, accorciando il test o magari spezzandolo in due tronconi (il programma è lungo circa 12K). L'indirizzo del club è Sinclair Club Roma - Talenti c/o Massimo D'ASCENZO via F. D'Ovidio, 109 Tel. 06/82.80.043 Buon test a tutti!

Test 2 - Siete zingari o pantofolai?
di Massimo D'Ascenzo
Sinclair Club Roma - Talenti

Test 2 - Siete zingari o pantofolai?
versione per ZX Spectrum 48K

```

102 PAPER 6: BORDER 4: INK 1: CLS
105 PRINT AT 6,3;"{5SG8} {3SG8}
   {3SG8} {5SG8}   {3SG8}"
108 PAUSE 30
111 PRINT AT 7,3;"   {SG8}   {SG8}
   {SG8}   {SG8}   {SG8}"
114 PAUSE 30
117 PRINT AT 8,3;"   {SG8}   {3SG8}
   {3SG8}   {SG8}   {SG8}   {3SG8}"
120 PAUSE 30
123 PRINT AT 9,3;"   {SG8}   {SG8}
   {SG8}   {SG8}   {SG8}"
126 PAUSE 30
129 PRINT AT 10,3;"  {SG8}   {3SG8}
   {3SG8}   {SG8}   {3SG8}"
132 PAUSE 30
137 PRINT AT 16,3;"SIETE ZINGARI O P
   ANTOFOLAI?"
138 PAUSE 30
139 PRINT AT 18,1;"by Massimo e Cinz
   ia D'Ascenzo": PAUSE 30
140 PRINT AT 19,1;"del SINCLAIR CLUB
   ROMA-TALENTI"
141 PAUSE 30
142 FOR Z=0 TO 21
144 PAPER 4: OVER 1: PRINT AT Z,0;"
   "

146 PAUSE 30
147 OVER 0
148 NEXT Z
150 CLS
1000 PRINT INVERSE 1;"SIETE ZINGARI
   O PANTOFOLAI"
1002 GO SUB 9980: PRINT
1010 PRINT "Vi sono alcune caratteris
   tiche della personalita' su cui
   crediamo valga la pena di
   soffermarci, non tanto pe
   r stabilire: si', sono cosi
   ', quanto per capire perche'
   siamo fatti cosi'."
1020 PRINT "Soltanto in questo modo i
   nfatti e' possibile, se lo desid
   eriamo, innescare un processo di
   cambiamento."
1030 PRINT "Il gioco-test che vi prop
   oniamo mette a confronto, ponend
   oli beninteso sullo stesso pi
   ano, due stili di vita."
1040 PRINT AT 21,0;"PREMI UN TASTO PE
   R CONTINUARE": PAUSE 0: CLS
1050 GO SUB 9950
1500 DIM a$(34)
2000 GO SUB 9982: GO SUB 8000
2010: PRINT " 1"
2020 PRINT "Dovrei decidermi a riordi
   nare lamia auto, perche' e' pien
   a di roba buttata sul sedile d
   i dietro, nel cofano, sotto
   i sedili."

```

```

2030 GO SUB 8201
2200 GO SUB 9984: GO SUB 8000
2210 PRINT " 2 "
2220 PRINT "La soluzione ideale per l
   e mie vacanze e' una seconda ca
   sa in un luogo che mi e' famili
   are e dove tornare tutti gli an
   ni."
2230 GO SUB 8200
2400 GO SUB 9986: GO SUB 8000
2410 PRINT " 3 "
2420 PRINT "Quando i soldi me lo perm
   ettono sono contento di rinnovar
   e l'arredamento della casa"

2430 GO SUB 8200
2600 GO SUB 9988: GO SUB 8000
2610 PRINT " 4 "
2620 PRINT "Adoro cenare in un rist
   orante cinese per assaggiare p
   iatti esotici"
2630 GO SUB 8200
2800 GO SUB 9980: GO SUB 8000
2810 PRINT " 5 "
2820 PRINT "Talvolta mi piace rilegge
   re un bel libro che ho gia' let
   to diversi anni or sono"
2830 GO SUB 8200
3000 GO SUB 9982: GO SUB 8000
3010 PRINT " 6 "
3020 PRINT "Sono affascinato all'idea
   di un viaggio in una regione se
   lvaggiae inesplorata"
3030 GO SUB 8200
3200 GO SUB 9984: GO SUB 8000
3210 PRINT " 7 "
3220 PRINT "Mi e' caro festeggiare il
   Natalemantenendo le tradizioni
   della mia infanzia"
3230 GO SUB 8200
3400 GO SUB 9986: GO SUB 8000
3410 PRINT " 8 "
3420 PRINT "Dedico con piacere il mio
   tempo libero ad arricchire e
   riordinare le mie collezi
   oni"
3430 GO SUB 8200
3600 GO SUB 9988: GO SUB 8000
3610 PRINT " 9 "
3620 PRINT "Mi spiace che le vecchie
   ricette della cucina familiare va
   dano perdute e ho cura di
   raccogliarle quando mi ca
   pita"
3630 GO SUB 8200
3800 GO SUB 9980: GO SUB 8000
3810 PRINT " 10 "
3820 PRINT "Se vincessi dieci milioni
   mi piacerebbe spenderli tutt
   i in una volta per una spesa u
   n po' pazza"

3830 GO SUB 8200
4000 GO SUB 9982: GO SUB 8000
4010 PRINT " 11 "
4020 PRINT "Sarei felice di trovare u

```

n lavoro che mi permettesse
di girare il mondo e venire
a contatto con ambienti sem
pre diversi"

4030 GO SUB 8200

4200 GO SUB 9984: GO SUB 8000

4210 PRINT " 12 "

4220 PRINT "Mi piacerebbe esercitare
una professione tramandata di
padre in figlio"

4230 GO SUB 8200

4400 GO SUB 9986: GO SUB 8000

4410 PRINT " 13 "

4420 PRINT "Metterei da parte un po'
di soldi per fare un viaggio
attraverso le capitali de
lla cultura europea"

4430 GO SUB 8200

4600 GO SUB 9988: GO SUB 8000

4610 PRINT " 14 "

4620 PRINT "Se c'e' una cosa che mi m
anca e' la diplomazia"

4630 GO SUB 8200

4800 GO SUB 9980: GO SUB 8000

4810 PRINT " 15 "

4820 PRINT "I mobili e gli oggetti di
casa mi creano un'atmosfera di
serenatranquillita' e tendo a
mantenerli sempre nella s
tessa disposizione"

4830 GO SUB 8200

5000 GO SUB 9982: GO SUB 8000

5010 PRINT " 16 "

5020 PRINT "Considero il trasloco com
e uno degli eventi piu' catastr
ofici che si possano abbattere
su una famiglia"

5030 GO SUB 8200

5200 GO SUB 9984: GO SUB 8000

5210 PRINT " 17 "

5220 PRINT "Fra le cose che ho piu' c
are vi e' certamente l'album del
le fotodi famiglia"

5230 GO SUB 8200

5400 GO SUB 9986: GO SUB 8000

5410 PRINT " 18 "

5420 PRINT "Non potrei mai separarmi
dai vecchi amici, anche se qu
alche volta sono un po' monoton
i"

5430 GO SUB 8200

5600 GO SUB 9988: GO SUB 8000

5610 PRINT " 19 "

5620 PRINT "Per me e' inconcepibile s
tipare una soffitta o una cantin
a di roba vecchia e inutile"

5630 GO SUB 8200

5800 GO SUB 9980: GO SUB 8000

5810 PRINT " 20 "

5820 PRINT "Trovo eccitante trascorre
re le vacanze in posti sempre d
iversi e con compagnie sempre nu
ove"

5830 GO SUB 8200

6000 GO SUB 9982: GO SUB 8000

6010 PRINT " 21 "

6020 PRINT "Ho cura della mia automob
ile e la cambio meno spesso dei
miei amici perche' mi dura a l
ungo in perfetta efficienza"

6030 GO SUB 8200

6200 GO SUB 9984: GO SUB 8000

6210 PRINT " 22 "

6220 PRINT "Detesto di tutto cuore le
riunioni in famiglia per
le feste"

6230 GO SUB 8200

6400 GO SUB 9986: GO SUB 8000

6410 PRINT " 23 "

6420 PRINT "Apro sempre accuratamente
i pacchi per conservare la
carta e lo spago"

6430 GO SUB 8200

6600 GO SUB 9988: GO SUB 8000

6610 PRINT " 24 "

6620 PRINT "Trovo stimolante cambiare
casa ed andare ad abitare in u
n altroquartiere o in un'altra c
itta'"

6630 GO SUB 8200

6800 GO SUB 9980: GO SUB 8000

6810 PRINT " 25 "

6820 PRINT "Sono convinto che ci si p
ossa innamorare piu' volte nel
la vita"

6830 GO SUB 8200

7000 GO SUB 9982: GO SUB 8000

7010 PRINT " 26 "

7020 PRINT "Mi darebbe un grande piac
ere occuparmi di restauri"

7030 GO SUB 8200

7200 GO SUB 9984: GO SUB 8000

7210 PRINT " 27 "

7220 PRINT "Se ne avessi le possibili
ta' mi piacerebbe provare a fare
del paracadutismo o volare co
n il deltaplano"

7230 GO SUB 8200

7300 GO SUB 9986: GO SUB 8000

7310 PRINT " 28 "

7320 PRINT "Il mezzo di trasporto che
vorreiusare sempre e' una jeep"

7330 GO SUB 8200

7400 GO SUB 9988: GO SUB 8000

7410 PRINT " 29 "

7420 PRINT "Mi piace scegliere accura
tamentela carta con la quale ric
oprire i libri ai quali tengo di
piu'"

7430 GO SUB 8200

7500 GO SUB 9980: GO SUB 8000

7510 PRINT " 30 "

7520 PRINT "Rinuncerei a un lavoro be
n renumerato e prestigioso
pur di non lasciare il mio paese
, la famiglia e gli amici"

7530 GO SUB 8200

7600 GO SUB 9982: GO SUB 8000

7610 PRINT " 31 "

7620 PRINT "Riconosco che la mia abit

```

azione e' piuttosto disordinata,
ma io ci vivo bene cosi'"
7630 GO SUB 8200
7700 GO SUB 9984: GO SUB 8000
7710 PRINT " 32 "
7720 PRINT "Se mi viene affidato un c
ompito del tutto nuovo per me mi
ci butto con entusiasmo e no
n mi lascio scoraggiare dalle
difficolta'"
7730 GO SUB 8200
7800 GO SUB 9986: GO SUB 8000
7810 PRINT " 33 "
7820 PRINT "Sostengo apertamente e co
n vigore opinioni politiche
diverse da quelle dei mie
genitori"
7830 GO SUB 8200
7900 GO SUB 9988: GO SUB 8000
7910 PRINT " 34 "
7920 PRINT "Se devo preparare un esam
e prendo appunti metodicame
nte, miorganizzo il tempo di stu
dio"
7930 GO SUB 8200
7940 GO TO 8240
8000 PRINT INVERSE !;" D O M A N
D A N. ";: RETURN
8200 NEXT x
8201 FOR x=1 TO 34
8205 INPUT "Vero/Falso (V/F) ? ";a$(x
)
8206 IF a$(x)<>"V" AND a$(x)<>"F"
THEN GO SUB 9950: GO TO 8205
8207 CLS
8215 RETURN
8240 GO SUB 9800
8250 IF a$(1)="V" THEN LET a=1
8251 IF a$(3)="V" THEN LET b=1
8252 IF a$(4)="V" THEN LET c=1
8253 IF a$(6)="V" THEN LET d=1
8254 IF a$(10)="V" THEN LET e=1
8255 IF a$(11)="V" THEN LET f=1
8256 IF a$(14)="V" THEN LET g=1
8257 IF a$(19)="V" THEN LET h=1
8258 IF a$(20)="V" THEN LET i=1
8259 IF a$(22)="V" THEN LET l=1
8260 IF a$(24)="V" THEN LET m=1
8261 IF a$(25)="V" THEN LET n=1
8262 IF a$(27)="V" THEN LET o=1
8263 IF a$(28)="V" THEN LET p=1
8264 IF a$(31)="V" THEN LET q=1
8265 IF a$(32)="V" THEN LET r=1
8266 IF a$(33)="V" THEN LET s=1
8270 LET y=a+b+c+d+e+f+g+h+i+l+m+n+o+
p+q+r+s
8278 GO SUB 9800
8280 IF a$(2)="F" THEN LET a=1
8281 IF a$(5)="F" THEN LET b=1
8282 IF a$(7)="F" THEN LET c=1
8283 IF a$(8)="F" THEN LET d=1
8284 IF a$(9)="F" THEN LET e=1
8285 IF a$(12)="F" THEN LET f=1
8286 IF a$(13)="F" THEN LET g=1
8287 IF a$(15)="F" THEN LET h=1

```

```

8288 IF a$(16)="F" THEN LET i=1
8289 IF a$(17)="F" THEN LET l=1
8290 IF a$(18)="F" THEN LET m=1
8291 IF a$(21)="F" THEN LET n=1
8292 IF a$(23)="F" THEN LET o=1
8293 IF a$(26)="F" THEN LET p=1
8294 IF a$(29)="F" THEN LET q=1
8295 IF a$(30)="F" THEN LET r=1
8296 IF a$(34)="F" THEN LET s=1
8300 LET w=a+b+c+d+e+f+g+h+i+l+m+n+o+
p+q+r+s
8302 LET k=y+w
8305 GO SUB 9800
8310 IF a$(1)="F" THEN LET a=1
8311 IF a$(3)="F" THEN LET b=1
8312 IF a$(4)="F" THEN LET c=1
8313 IF a$(6)="F" THEN LET d=1
8314 IF a$(10)="F" THEN LET e=1
8315 IF a$(11)="F" THEN LET f=1
8316 IF a$(14)="F" THEN LET g=1
8317 IF a$(19)="F" THEN LET h=1
8318 IF a$(20)="F" THEN LET i=1
8319 IF a$(22)="F" THEN LET l=1
8320 IF a$(24)="F" THEN LET m=1
8321 IF a$(25)="F" THEN LET n=1
8322 IF a$(27)="F" THEN LET o=1
8323 IF a$(28)="F" THEN LET p=1
8324 IF a$(31)="F" THEN LET q=1
8325 IF a$(32)="F" THEN LET r=1
8326 IF a$(33)="F" THEN LET s=1
8328 LET j=a+b+c+d+e+f+g+h+i+l+m+n+o+
p+q+r+s
8330 GO SUB 9800
8335 IF a$(2)="V" THEN LET a=1
8336 IF a$(5)="V" THEN LET b=1
8337 IF a$(7)="V" THEN LET c=1
8338 IF a$(8)="V" THEN LET d=1
8339 IF a$(9)="V" THEN LET e=1
8340 IF a$(12)="V" THEN LET f=1
8341 IF a$(13)="V" THEN LET g=1
8342 IF a$(15)="V" THEN LET h=1
8343 IF a$(16)="V" THEN LET i=1
8344 IF a$(17)="V" THEN LET l=1
8345 IF a$(18)="V" THEN LET m=1
8346 IF a$(21)="V" THEN LET n=1
8347 IF a$(23)="V" THEN LET o=1
8348 IF a$(26)="V" THEN LET p=1
8349 IF a$(29)="V" THEN LET q=1
8350 IF a$(30)="V" THEN LET r=1
8351 IF a$(34)="V" THEN LET s=1
8354 LET t=a+b+c+d+e+f+g+h+i+l+m+n+o+
p+q+r+s
8357 LET u=j+t
8360 GO SUB 9980: PRINT AT 10,5;"U N
M O M E N T O"
8362 PAUSE 50
8364 PRINT AT 16,2;"Sto elaborando i
risultati"
8366 PAUSE 150: CLS
8367 IF k<23 AND u<22 THEN GO TO 920
0
8368 GO SUB 9982: PRINT AT 8,3;"IL PU
NTEGGIO DA TE OTTENUTO"
8369 PRINT AT 14,8;"e' di punti ";
8370 IF k>=29 THEN GO TO 8400

```

```

8375 IF k>=23 AND k<=28 THEN GO TO 8
      600
8380 IF u>=29 THEN GO TO 8800
8385 IF u>=23 AND u<=28 THEN GO TO 9
      000
8400 PRINT k: PAUSE 100: CLS
8402 GO SUB 9984: PRINT INVERSE 1;"S
      IETE UNO ZINGARO": PAUSE 100
8404 PRINT
8406 GO SUB 9850
8410 PRINT "Sei veramente uno zingaro
      , si direbbe che non hai ne' p
      atri a ne' casa. Non metti radic
      i in nessun luogo, sei attratt
      o dalleavventure e rifiuti quals
      iasi legame."
8420 GO TO 9790
8600 PRINT k: PAUSE 100: CLS
8602 GO SUB 9986: PRINT INVERSE 1;"S
      IETE UNO ZINGARO": PAUSE 100
8606 GO SUB 9850
8610 GO SUB 9986: PRINT "Sei un prima
      rio deciso rivolto verso il dom
      ani, capace di iniziative c
      he generalmente porti a term
      ine, affronti il domani e l'i
      mprevisto senza timore"
8620 GO TO 9790
8800 PRINT u: PAUSE 100: CLS
8802 GO SUB 9988: PRINT INVERSE 1;"S
      IETE UN PANTOFOLAIO": PAUSE 100
8804 PRINT
8806 GO SUB 9900
8810 PRINT "Sei un pantofolaio, non
      abbandoni i luoghi e le p
      ersone care per nessuna ragione
      al mondo. Le novita' ti turb
      ano, rimpiangi continuamente i
      l buon tempo antico. Sei scrupol
      oso e monogamo."
8820 GO TO 9790
9000 PRINT u: PAUSE 100: CLS
9002 GO SUB 9988: PRINT INVERSE 1;"S
      IETE UN PANTOFOLAIO": PAUSE 100
9004 PRINT
9006 GO SUB 9900
9010 GO SUB 9980: PRINT "Ami le tradi
      zioni, le comodita' della casa,
      la cultura, sei metodico e a
      ffidabile nel lavoro, hai
      cura delle tue cose, coltivi la p
      arentela e le amicizie."
9020 GO TO 9790
9210 GO SUB 9982: PRINT "Sei in una b
      uona situazione di equilibrio.
      Le tradizioni sono forti, ma ti
      danno sicurezza per cercare nuov
      e strade. Non prenderesti
      l'astronave per AlfaCentauri, ma
      un viaggio avventuroso
      e pieno di imprevisti n
      on ti dispiacerebbe."
9790 STOP
9800 LET a=0: LET b=0: LET c=0: LET d
      =0: LET e=0: LET f=0: LET g=0:

```

```

      LET h=0: LET i=0: LET l=0:
      LET m=0: LET n=0: LET o=0:
      LET p=0: LET q=0: LET r=0:
      LET s=0: RETURN
9850 GO SUB 9984: PRINT "Sei una pers
      ona che si adatta facilmente a
      l presente, piu' pronta ad ag
      ire che a riflettere,
      ti adatti immediatamente
      te a circostanze impreviste,
      reagisci senza ansia;"
9852 PRINT "sei una persona impulsiv
      e, ma non tieni rancore, non sei
      costante ne' fedele e non tieni
      nemmeno ai ricordi del passato."
9854 PRINT AT 19,0;"PREMI UN TASTO PE
      R CONTINUARE": PAUSE 0: CLS
9856 RETURN
9900 GO SUB 9986: PRINT "Sei una pers
      ona che ha bisogno di una forte
      spinta per mettersi in moto, hai
      difficolta' ad adattarti al
      l'imprevisto ed ai cambiamenti,
      "
9902 PRINT "preferisci le cose note,
      familiari, hai bisogno di
      un tempo di riflessione prim
      a di agire, hai radici molto f
      orti nel passato."
9904 PRINT AT 19,0;"PREMI UN TASTO PE
      R CONTINUARE": PAUSE 0: CLS
9905 RETURN
9906 GO SUB 9900
9950 CLS : PRINT INVERSE 1;" I M
      P O R T A N T E"
9953 PRINT
9956 PRINT "Ad ogni singola domanda d
      ire se secondo voi
      l'affermazione e' VERA op
      pure FALSA"
9960 PRINT
9965 PRINT "Se volete rispondere VERO
      dovetepremere il tasto V (solo
      nella posizione di maiuscolo)"
9968 PRINT
9970 PRINT "Se volete rispondere FALS
      O dovette premere il tasto F
      (solo nella posizione di maiusc
      olo)"
9975 PAUSE 300: CLS : RETURN
9980 PAPER 6: BORDER 6: INK 0: CLS :
      RETURN
9982 PAPER 4: BORDER 4: INK 0: CLS :
      RETURN
9984 PAPER 7: BORDER 7: INK 1: CLS :
      RETURN
9986 PAPER 3: BORDER 3: INK 7: CLS :
      RETURN
9988 PAPER 5: BORDER 5: INK 0: CLS :
      RETURN

```


input·output

**Piccoli
annunci**

Vendo programmi come Pac Man, Space Invaders, Othello (premio internazionale), utilities varie. Inviare richieste. Programmi sotto L. 2.500 circa. Inoltre dispongo di cartridge "Gorf" per VIC 20. Vincenzo Mascetti - Via Roma, 8 - 22077 Olgiate Comasco (CO) - Tel. 944022

Vendo VIC 20 nuovo, ancora imballato più espansione originale 8 Kbyte, manuali vari in italiano, 80 giochi su cassette originali (non copiate). Il tutto a solo L. 150.000. Piero Discacciati - Via Paganini, 28/B - 0052 Monza - Tel. 039/329412

Vendo giochi ed utility per Commodore 64. Paolo Clerly - Via Ponti Rossi, 37 - 80131 Napoli - Tel. 081/7413749

Per Commodore 64 vendo: cassetta con 15 programmi Totocalcio e Totip a L. 50.000; cassetta con 11 programmi di musica a L. 40.000. Invio lista con 800 programmi. Eliseo Mastrangelo - Via Casilina, 1641 - 00133 Roma - Tel. 06/6151345

Solo con privati cambio e compro a prezzi contenuti qualsiasi tipo di programma per VIC 20 non espanso (listati e nastri). Cerco inoltre un manuale sulla programmazione in linguaggio macchina per VIC 20. Enrico Rasero - Via Padova, 276 - 20132 Milano - Tel. 02/2567000

Compro espansione da 16 Kbyte RAM per VIC 20. Telefonare ore 19-21. Pietro Saronni - Piazza De Gasperi, 7 - 20064 Gorgonzola - Tel. 02/9510208

Per C64 vendo 10 giochi tra i migliori (Calcio, Decathlon, Biliardo, Scacchi, Donkey Kong, Seawolf, Squish'em, Wizard Of War, Frogger, Pac Man). Tutti in cassetta a L. 50.000. Giovanni Cricchio - Via Gen Corselli, 16 - 90123 Palermo - Tel. 473680

Vendo, cambio fantastici programmi su nastro tutti in Turbo Tape a L. 20.000: Decathlon, Basket, Dig Dug, Popeye, Jungle Hunt, Pole Position, Pengo, QBert, Zaxxon, Congo Bongo ed altri 100. Massima serietà. Etienne Caldironi - Via Bargigia - 48100 Ravenna - Tel. 0544/39619

Vendo per VIC 20 cassetta contenente giochi (da un minimo di 5 ad un massimo di 10) tutti in LM e privi di espansione tra quelli qui elencati: Space Invader, Blitz, Scramble, Carneval, Luna Park, Centipede, Laser, Pac Man, Wall, Snake Pit. Vendo a L. 2.500 l'uno comprese spese postali, a chi acquisterà 10 giochi in regalo Briscola e Sette e 1/2. Telefonare dalle 15 alle 20. Salvo D'Urso - Via La Farina, 3B - 95018 Riposto (CT) - Tel. 095/935075

Vendiamo o scambiamo programmi su cassetta o su disco per C64. Preferibilmente zona Torino. Fabrizio Vitagliano - Via Servais, 73 - 10146 Torino - Tel. 011/720666

Per C64 vendo, cambio tanti giochi ed utility. Simons' BASIC L. 5.000, giochi L. 1.000. Vendo inoltre Grand Master a L. 2.500. Inviare la vostra lista per cambio. Massima serietà. Adriano Belletti - Via Belvedere, 18 - 40069 Zola Predosa (BO) - Tel. 051/751203

Vendo VIC 20 più registratore "Maxtron", joystick Spectravideo Quick Shot II, 4 cartridges (Sargon II, Raid On Fort Knox, Gorf, Radar Rat Race) più 2 libri ("VIC 20 User Guide", "Alla scoperta del VIC 20"), 55 programmi su cassetta a L. 380.000 non trattabili. Francesco Guelfi - Via Zaninelli, 7 - 26013 Crema (CR) - Tel. 86798

Vendo per VIC 20 giochi, da scegliere su una lista di 100, a L. 1.500 l'uno. Il primo che ordina 20 giochi riceverà un libro in regalo. Per ricevere la lista inviare L. 1.000 e scrivere a: Nicola Simone - Corso Cincinnato, 224 - 10153 Torino

Siamo due amici che hanno fondato un club del computer. Chi volesse aderire o comprare dei giochi per VIC 20 telefoni ad Alessandro allo 02/3566087 oppure a: Alessandro Rossi - Via Baranzate, 57 - 20026 Novate Milanese - Tel. 02/3564203

Vendo, cambio 150 programmi su nastro di giochi ed utility per C64. Alberto Albertini - Via N. Bixio, 197 - 37069 Villafranca (VR) - Tel. 045/7900152

Vendo, compro, cambio programmi per Commodore 64 (oltre 500) sia su disco che su nastro. Inviare lista e risponderò con la mia, annuncio sempre valido. Scrivere o telefonare a: Gianni Mazzesi - Via Cella, 329 - 48020 S. Stefano (RA) - Tel. 573520

Vendo causa passaggio a CBM 64 VIC 20 con garanzia (5 mesi), più di 100 giochi su cassetta a L. 200.000, espansione 32 Kbyte a L. 160.000. Tutto a L. 300.000. Cristian Megazze - Via Stelvio, 22 - 20025 Legnano (MI) - Tel. 549115

Commodoriani attenzione! Si vendono cassette giochi per Commodore 64 ad ottimo prezzo. Disponiamo anche di una vasta gamma di giochi su floppy disk. Per avere la lista inviare L. 1.000. Massimo Campili - Via Carnia, 8/A - 05100 Terni - Tel. 0744/59497

Vendo per VIC 20 3 cartucce: Male Attack, Cosmic Cruncher, Invasion a L. 90.000 ciascuna! Vendo inoltre joystick Triga a grilletto a L. 15.000. Compro espansione 16 Kbyte a basso prezzo. Gianluca Gaetano - Via Fleming, 2 - 89026 S. Ferdinando (RC)

Vendo a L. 4.000 fantastici giochi per C64 come: Pit Stop, Hunch Bach, Pole Position, Parà Truppen, Le Mans, ecc. Telefonare a: Mario Festinese - Via Giuseppe Capaldo, 5 - 80128 Napoli - Tel. 364303

Vendo programmi per VIC 20 e CBM 64 in cassetta o soltanto listati. Per ottenere il depliant spedire L. 1.000 in busta chiusa a: Fabio Molinari - Via Gerolamo Belloni, 112 - 00191 Roma

Strordinario! Vendo per CBM 64 cassetta con Turbo Tape, International Soccer, Pole Position, Flipper, Pengo, One On One (Pallacanestro), Biliardo, Dig Dug, Space Invaders, Moon Buggy, Hamburger Time, Donkey Kong, Puc Man. Il tutto a L. 20.000. Cesare Malara - Via Napoleona, 12 - 22100 Como - Tel. 031/267327

Vendo a prezzi incredibili programmi per VIC 20 in versione base (funzioni, programmi, ecc.). Per notizie scrivere a: Paolo Detti - Via Calabria, 19 - 58100 Grosseto - Tel. 0564/413313

Compro programmi utility per VIC 20 con espansione 16 Kbyte; cerco in particolare programmi per statistica. Vincenzo Mani - Contrada Ranieri - 87030 Roges (CS) - Tel. 0984/838768

Per Commodore 64 vendo i seguenti giochi su cassetta: Purple Turtles, Calcio, QBert, Juice, Basket, Flight Path 737, Lazy Jones, Beach Head, Decathlon a L. 50.000. Filippo Leto - Via S. D'Acquisto, 2 - 21100 Varese - Tel. 0332/230790

Vendo ZX Spectrum 80 Kbyte con BASIC esteso e oltre 500 programmi con tutte le istruzioni. Guido D'Elia - Via Pieve Di Cadore - 00135 Roma - Tel. 06/3387408

Vendo VIC 20 più registratore C2N, 4 cartucce: Slot, Radar Ratrice, Avenger, Scacchi; joystick, alta risoluzione, 3 cassette di programmi e giochi, il libro del VIC e 2 di programmazione. Tutto L. 360.000. Bevatì Umberto - Via B. Sangalli, 1 - 20052 Monza (MI) - Tel. 039/362283

Vendo e cambio giochi ed utility esclusivi a prezzi eccezionali: Simons' BASIC a L. 10.000; Magic Desk I-II, Koala Painter, Mission Impossible L. 20.000 in Lm; S.A.M. (il fantastico programma parlante); Arabian Nights, e tantissimi altri. Per informazioni basterà scrivere o telefonare dopo le 21 a: Ezio De Ritis - Via Giulio Sigismondi, 11 - 66034 Lanciano (CH) - Tel. 0872/49067

Vendo programmi per C64 come Decathlon, Indiana Jones, ecc; i prezzi vanno dalle L. 500 alle L. 2.500. Richiedere lista e prezzi a: Marco Romani - Via G. Valli, 95 - 00149 Roma - Tel. 06/5265121

Vendo VIC 20 più registratore e 60 giochi a L. 200.000. Roberto Leo - Via Delle Forze Armate, 66 - 20147 Milano - Tel. 02/4034233

Compro, vendo, cambio programmi e videogames per il C64; programmi e videogames seri per persone valide. Livio Modena - Via Dante, 110 - 18038 Sanremo - Tel. 0184/73079

Vendo VIC 20 usato poco più registratore, espansione 3-8-16 Kbyte, joystick, 250 giochi, molti in linguaggio macchina (Bonzo, Boss, Defenda, Arcadia e utility) a L. 400.000 trattabili. Per informazioni telefonare a: Paolo Pedrona - Via Trieste, 38 - 43100 Parma - Tel. 0521/75597

Vendo il libro "Alla scoperta del VIC" a sole L. 11.000. Inoltre vendo programmi per C64 (Decathlon, Mondial Soccer, ecc) a L. 3.000 cadauno oppure cassetta (50 giochi) a L. 25.000. Salvatore Lenza - Via C. Mediterraneo, 34 - 87029 Scalea (CS) - Tel. 0985/20177

Svendo a L. 4.000 cadauno oltre 150 giochi per C64. Cerco inoltre materiale circa il linguaggio macchina. Salvatore Lenza - Via C. Mediterraneo, 34 - 87029 Scalea (CS) - Tel. 0985/20177

Cambio software per Commodore 64 (solo zona Milano). Marco Scorta - Via San Francesco, 13 - 20091 Bresso (MI) - Tel. 02/610061

Vendo per C64 oltre 50 programmi, tutti in LM, tra cui (Suicide Strike, Fort Apocalypse, Atzec, Hunckback, Scramble, ecc.) a L. 2.000 l'uno. Scrivere a: Saverio Managò - Via M. Suello, 2 - 36100 Vicenza - Tel. 044/30175

Spedisco una cassetta con 50 giochi per C64 a chi mi manda una sua cassetta con altrettanti programmi e giochi. Massima serietà. Enrico Pacchini - Via Caldanelle, 4 - 57029 Venturina (LI) - Tel. 0565/850436

Vendo VIC 20 in ottime condizioni più espansione 16 Kbyte, 50 giochi tra cui: Sette e mezzo, Briscola, Totocalcio, ecc ... Il tutto a L. 300.000. Telefonare ore pasti o scrivere a: Salvatore Turino - Via Alongi, 49 - 95040 Mirabella Imbaccari (CT) - Tel. 0933/991148

Vendo, cambio a L. 1.000 centinaia di giochi originali per C64 tra i quali: Pitfall I e II, King Kong, International Soccer, Jungle Hunt, Decathlon, Poeye, Basket, Zaxxon e tanti altri. Massima serietà, telefonare o scrivere dalle 14 alle 17 e dalle 21 alle 22. Carlo Di Tullio - Via Da Denominarsi, 5 - 71037 Monte S. Angelo (FG) - Tel. 0884/62589

Vendo per Commodore 64 programmi gestionali o giochi su cassetta, da L. 2.000 a L. 9.000. Scrivete o telefonate a: Andrea Magro - Via Ercole Ferrario, 6 - 20144 Milano - Tel. 02/435419

Cambio, vendo per C64 50 games, tutti in LM con Turbo Tape. Invio lista a chi mi scrive. Cambio per Karate e Break Dance: Skramble, Shamus, Brands, Pole Position 1, B.C., Fort Apocalypse. Inviare la vostra lista. Salvatore Strangio - Via Redipuglia - 89013 Gioia Tauro (RC)

SUPERSING

CEOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.sofr - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

LIBRI

Al confini dello Spectrum -
Applicazioni avanzate

cod. AJACO46 L. 25.200

Il Basic in 30 ore
per lo Spectrum

cod. AJAK027 L. 36.000

Ordino i seguenti libri per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

Cod. Cod.

Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp n° 19445204 intestato a J.sofr - Milano

Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i
soci in Italia)

Nome

Cognome

Via

Città

Se richiesta fattura:
Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSING INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a
tutti i lettori. Chi desidera comprare, vendere o cam-
biare hardware o software può inviare il tagliando
a J.sofr - Viale Restelli, 5 - 20124 MILANO

COMPRO VENDO CAMBIO

ZX80 ZX81 ZX Spectrum QL SOFTWARE

PERIF.

Nome Cognome
Via C.A.P.
Città Tel.

**SUPERSING è bello, però... (ovvero suggerimenti,
idee, critiche, richieste e tutto ciò che vi passa per
la testa).**

Nome Cognome
Via C.A.P.
Città Tel.

SEIKOSHA



NON AVRAI ALTRA STAMPANTE

Seikosha ti invita nel meraviglioso mondo delle sue stampanti.

Un mondo fatto di progresso, di elevatissima qualità, velocità e silenziosità di stampa.

Seikosha oggi ti propone la più vasta gamma di stampanti nate per esaltare le prestazioni di ogni tipo di computer.

All'altezza di ogni esigenza, anche della tua che usi i Personal Computer Sinclair.

Piccola e compatta, dalle prestazioni generose, GP 50 S con 35 caratteri al secondo e 32 colonne, è la stampante ideale per risolvere con soddisfazione le prime esigenze di stampa di chi usa lo ZX Spectrum.

Se possiedi anche l'interfaccia 1, niente di meglio della stampante GP 500 S con 50 caratteri al secondo e 80

colonne che ti consente utilizzi anche di tipo gestionale.

Se lavori con un computer Sinclair QL, non puoi rinunciare agli 80 caratteri al secondo e 80 colonne anche Near Letter Quality a 20 caratteri per secondo del modello SP 800 IQL.

Se poi le tue esigenze sono altamente professionali, la stampante BP 5420 A con 136 colonne, 420 caratteri al secondo, anche Near Letter Quality a 104 caratteri al secondo, rende ancor più grande il tuo Sinclair QL.

Seikosha e Sinclair: una coppia che va d'amore e d'accordo.

SEIKOSHA

Distribuzione esclusiva: GBC Divisione Rebit



32 BIT

32 BIT

16 BIT

16 BIT

8 BIT

8 BIT



SINCLAIR QL: AL VERTICE DELLA NUOVA GENERAZIONE

Sinclair QL rivoluziona il mondo dei computer, perché combina le dimensioni di un home con la potenza e la capacità di un mini.

QL è l'unico computer, nella sua fascia, ad impiegare il microprocessore a 32 bit, quando gli altri si fermano a 8 oppure 16.

La sua portentosa memoria è di 128 KRAM espandibile a 640.

I quattro programmi applicativi, già incorporati, sono immediatamente utilizzabili e superano, in qualità, il software dei microcomputer esistenti.

Ha la possibilità di multitask e può essere inserito in reti di comunicazione.

Grazie ai due microdrive e al software incorporati, Sinclair QL, nella sua confezione originale, è già pronto per l'uso: basta collegarlo ad un video.

E pensare che tutta questa tecnologia pesa meno di due chili e trova spazio in una normale 24 ore.

Un computer così non poteva che essere Sinclair.

sinclair

Distribuzione esclusiva: GBC Divisione Rebit.

Tutti i prodotti Sinclair, distribuiti da GBC Divisione Rebit, sono corredati da regolare certificato di garanzia italiana.